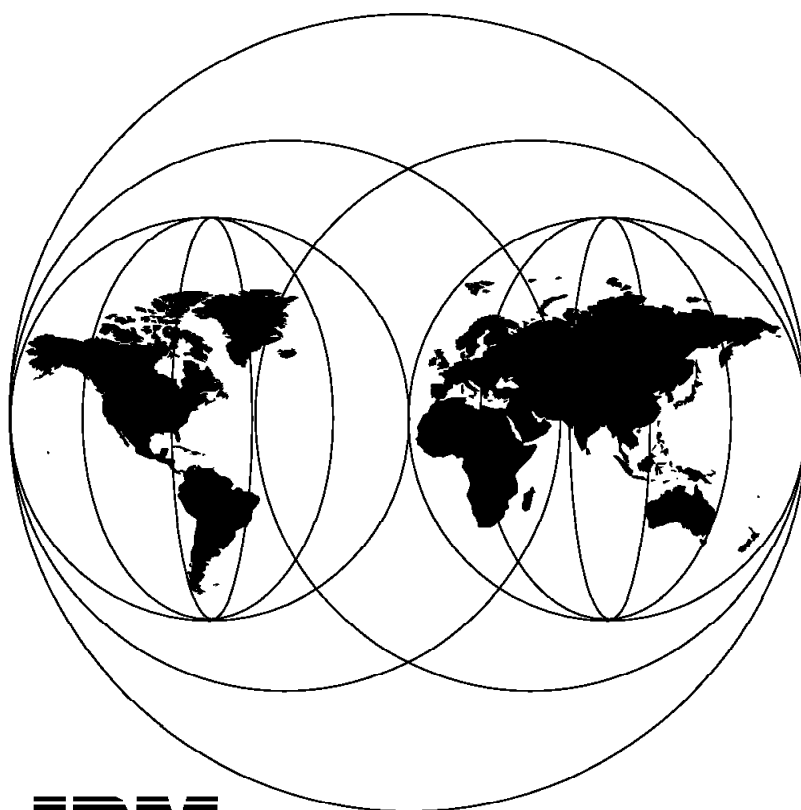


VisualAge Generator Client/Server Communications Examining the Options

May 1997



**International Technical Support Organization
San Jose Center**



International Technical Support Organization

SG24-4237-00

**VisualAge Generator Client/Server Communications
Examining the Options**

May 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, "Special Notices" on page 173.

First Edition (May 1997)

This edition applies to:

- Version 2.2 of IBM VisualAge Generator Developer for OS/2, Program 5622-580 for use with the OS/2 Operating System
- Version 2.2 of IBM VisualAge Generator Workgroup Services for OS/2, AIX, and Windows, Program 5622-585 for use with the OS/2, AIX, and Windows Operating Systems

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
Tables	ix
Preface	xi
The Team That Wrote This Redbook	xi
Comments Welcome	xii
Chapter 1. Introduction	1
1.1 Project Scope and Objectives	2
1.2 Positioning VisualAge Generator Client/Server in the Open Blueprint	3
1.2.1 Open Blueprint Components	4
1.2.2 Critical Services	5
1.2.3 Integration Objectives	7
1.3 VisualAge Generator Communication Services Capabilities	8
1.3.1 CICS	9
1.3.2 Distributed Computing Environment	12
1.3.3 VisualAge Generator Middleware	13
1.3.4 Summary	16
1.4 Implementing Client/Server Systems with VisualAge Generator	16
Chapter 2. Testing Applications with VisualAge Generator Developer ITF	19
2.1 Calling Generated Applications	19
2.1.1 Runtime Processing of Remote Calls	20
2.1.2 ITF Processing of External Calls	20
2.2 Database Identification and Authorization	21
2.2.1 ITF Database Identification	21
2.2.2 ITF Database Authorization	22
2.2.3 Runtime Database Identification	22
2.2.4 Runtime Database Authorization	23
Chapter 3. CICS-Based Client/Server	25
3.1 CICS Configuration	25
3.1.1 Options	25
3.1.2 Protocol Choices	26
3.1.3 Processing Flow	27
3.2 CICS Client/Server Scenarios	28
3.2.1 CICS OS/2 Server	29
3.2.2 CICS/6000 Server	39
3.2.3 CICS Client	44
3.2.4 CICS-to-CICS Connections	49
Chapter 4. DCE-Based Client/Server	53
4.1 DCE Configuration	53
4.1.1 Options	53
4.1.2 Protocol Choices	54
4.1.3 DCE-Based Client/Server Communication Processing	55
4.2 DCE Client/Server Scenarios	56
4.2.1 Configuration Basics	56

4.2.2 AIX DCE Cell Server Workstation	59
4.2.3 OS/2 DCE Application Server Workstation	65
4.2.4 OS/2 DCE GUI Client Workstation	68
4.2.5 Windows NT DCE Application Server Workstation	70
4.2.6 Windows 95 DCE GUI Client Workstation	73
4.2.7 Running Applications with DCE-Based Client/Server Communication	75
4.2.8 Implementing Secure DCE-Based Client/Server Communication	88
4.2.9 Common DCE-Based Client/Server Communication Configuration Error	98
Chapter 5. Database-Enabled Client/Server	103
5.1 Database Configuration Options	104
5.1.1 DB2 Common Server Connectivity	104
5.1.2 Distributed Relational Database Architecture Connectivity	104
5.1.3 DataJoiner	105
5.2 Accessing Relational Databases with VisualAge Generator	106
5.2.1 Configuring Database Support	106
5.2.2 Implementing Distributed Unit of Work Database Access	111
5.2.3 Sample DUOW Application Configurations	116
Chapter 6. DBCS-Enabled Client/Server Configurations	139
6.1 DBCS Overview	139
6.1.1 Starting BookManager in a DBCS Environment	140
6.1.2 DBCS Data Definition	140
6.1.3 Relational Database Considerations	141
6.2 Client/Server Implementation for DBCS Applications	141
6.2.1 DBCS Client/Server Scenarios	141
6.2.2 Environment Setup and Customization	145
Appendix A. Sample Applications	163
A.1 VGPING Sample Application	163
A.1.1 Objectives	163
A.1.2 Overview	164
A.1.3 Function	165
A.1.4 Sample Application Materials	167
A.2 DUOW Sample Application	168
A.2.1 Objectives	168
A.2.2 Overview	168
A.2.3 Function	169
A.2.4 Sample Application Materials	171
Appendix B. Special Notices	173
Appendix C. Related Publications	175
C.1 International Technical Support Organization Publications	175
C.2 Redbooks on CD-ROMs	175
C.3 Other Publications	176
How to Get ITSO Redbooks	177
How IBM Employees Can Get ITSO Redbooks	177
How Customers Can Get ITSO Redbooks	179
IBM Redbook Order Form	180
List of Abbreviations	181
Index	183
ITSO Redbook Evaluation	185

Figures

1.	VisualAge Generator Client/Server Implementation Options	1
2.	VisualAge Generator Client/Server Configuration Options	2
3.	Components of the Open Blueprint	4
4.	VisualAge Generator Client/Server and Distributed Computing Options	8
5.	CICS Client/Server Configurations	26
6.	CICS Client/Server Communication Processing Flow	27
7.	CICS Configuration Implemented during Residency Project	29
8.	CICS OS/2 SIT—Page 1	32
9.	CICS OS/2 SIT—Page 2	32
10.	CICS OS/2 SIT—Page 3	33
11.	Control Logic in ELAENV.COMD to Limit to One Pass Execution	33
12.	Control Settings in ELAENV.COMD	34
13.	Linkage Table for CICS OS/2 Sample Application Servers	37
14.	Generate Statements for CICS OS/2 Sample Application Servers	38
15.	Software Installed to Support CICS/6000 Server	39
16.	Linkage Table for CICS/6000 Sample Application Servers	43
17.	Generate Statements for CICS OS/2 Sample Application Servers	43
18.	VisualAge Generator Script to Add Application Definitions to CICS/6000	44
19.	Sample Application Generation Commands for GUI Client	45
20.	Linkage Table used to Generate Sample Application GUI Client for CICS Servers	46
21.	CICSCLI.INI Definitions for Workstation CICS Servers	47
22.	Linkage Table used for Runtime GUI Client Calls to CICS Servers	48
23.	CSOTRACE.OUT Entries for GUI Client Call to CICS OS/2 Server	48
24.	CICS/6000 Communications Identifier for CICS OS/2 System	50
25.	CICS/6000 Program Definition for Remote CICS OS/2 Program	50
26.	VGPING Sample Application GUI after Three-Tier Call: GUI to CICS/6000 to CICS OS/2	51
27.	CICS OS/2 Communications Identifier for CICS/6000 System	51
28.	CICS OS/2 Program Definition for Remote CICS/6000 Program	52
29.	VGPING Sample Application GUI after Three-Tier Call: GUI to CICS OS/2 to CICS/6000	52
30.	DCE-Based Client/Server Configurations	54
31.	Distributed Operation in a DCE Cell	55
32.	VisualAge Generator Client/Server Communication DCE Configuration Overview	58
33.	PING Command Used to Test TCP/IP Communication between Workstations	58
34.	Defining DCE Organizations, Groups, and Users on AIX Cell Server with dcecp	61
35.	Defining Base DCE Directories on AIX Cell Server with dcecp	61
36.	OBJECT ACL for VisualAge Generator DCE Base Directory	62
37.	Adding Application Server Principal to CDS Administrator Group	62
38.	Adding Authority for Base Directory to Application Server Group	63
39.	Removing Test Authority on Base Directory for unauthenticated and any_other ACL Entries	64
40.	Creating Target DCE Directory on AIX Cell Server with dcecp	65
41.	OBJECT ACL List for VisualAge Generator DCE Target Directory	65
42.	Linkage Table for OS/2 Sample Application Servers with DCE-Based Client/Server Communication Support	67

43.	Generating OS/2 Sample Application Servers with DCE-Based Client/Server Communication Support	67
44.	Linkage Table for OS/2 Sample Application Client with DCE-Based Client/Server Communication Support	69
45.	Sample Application Generation Commands for OS/2 Client for DCE-Based Client/Server Communication	70
46.	Linkage Table for Windows NT Sample Application Servers with DCE-Based Client/Server Communication Support	72
47.	Generating Windows NT Sample Application Servers with DCE-Based Client/Server Communication Support	72
48.	Linkage Table for Windows 95 Sample Application Client with DCE-Based Client/Server Communication Support	74
49.	Sample Application Generation Commands for Windows 95 GUI Client for DCE-Based Client/Server Communication	74
50.	VisualAge Generator DCE-Based Client and Application Server Configuration	75
51.	DCE.CNF Configuration File for OS/2 DCE Application Server	76
52.	DCE.CNF Configuration File for Windows NT DCE Application Server	76
53.	DCE.CNF Configuration File for AIX DCE Application Server	77
54.	Adding Key Table Entry for DCE Application Server Principal	77
55.	CDS Directories for SERVERID on AIX Cell Server	78
56.	Display for a Running CSODCES DCE Server Program	79
57.	CDS Directories and Application Objects on AIX Cell Server	79
58.	Information for a CDS Object as Shown Using cdsdp	80
59.	Application Objects for OS/2, Windows NT, and AIX DCE Application Server Workstations	80
60.	VisualAge Generator Windows 95 Client Trace Log for DCE-Based Client/Server Communication Calls	81
61.	DCE-Based Client/Server Communication Call Path Options	83
62.	Primary DCE Server Program Startup Messages	85
63.	Secondary DCE Server Program Startup Messages	85
64.	CDS Object Information for Two Application Server Workstations	86
65.	Client/Server Communication Trace Log for Multiple Application Server Workstations	86
66.	Secondary DCE Server Program Shutdown Messages	87
67.	Primary DCE Server Program Shutdown Messages	88
68.	Checking Existing ACL for CDS Directory with acl_edit	89
69.	Initial ACL Entry for LOCATION Object	90
70.	acl_edit Commands to Remove Unauthenticated User Access	90
71.	cdscp Commands to List and Delete SERVERID Directory Objects	90
72.	ACL Entry List for Location Object to Force DCE Logon	91
73.	Testing Directory Access for Unauthenticated DCE Users with acl_edit Commands	91
74.	Trace Log for Unauthenticated VisualAge Generator Server Application Call	92
75.	Testing Directory Access for Authenticated DCE Users with acl_edit Commands	92
76.	Trace Log for Authenticated VisualAge Generator Server Application Call	93
77.	acl_edit Commands to Permit Test Access for vgusradm Group	94
78.	cdscp Commands to Delete LOCATION Object and then List Remaining SERVERID Directory Objects	95
79.	Linkage Table for OS/2 Sample Application Servers with Secure DCE-Based Client/Server Communication Support	95
80.	Secure DCE Server Program Startup Messages	96
81.	ACL Entry List for Location Object with vgusradm Group	96
82.	Trace Log for Rejected Authenticated VisualAge Generator Server Application Call	97
83.	Trace Log for Accepted Authenticated VisualAge Generator Server Application Call	97
84.	DB2 Common Server Configurations	104
85.	DB2 DRDA Configurations	105
86.	DataJoiner Support for Access to Non-IBM Databases	106
87.	Trace of Implicit EZERSQLDB Database Connection and Successful Database Access	109

88.	Trace of Implicit DB2DBDFT Database Connection and Database Access	110
89.	Trace of Implicit Database Connection and Database Access Failure	110
90.	VisualAge Generator EZECONCT Statement Implementation for DUOW Support	112
91.	DB2/2 Common Server Configuration	117
92.	Distributed Database Nodes and Cataloged Databases	119
93.	Remote DB2/2 and DB2/MVS Database Configuration	120
94.	Communications Manager Network Definition File for DRDA Connection	121
95.	Staff Table Definitions for Distributed Database Environment	124
96.	Linkage Table for DUOW Called Application	126
97.	Generate Statements for DUOW Sample Application	126
98.	DUOW Sample Application Database Connection Processing Logic	126
99.	Standard Testing Cycle for DUOW Configurations	127
100.	Local Database Connection and Test SELECT Statements	128
101.	FCWTRACE.OUT File for Distributed Access to Local Databases	129
102.	FCWTRACE.OUT File for Distributed Access to Local and Remote Databases	132
103.	DUOW Configuration with Two Remote DB2/2 Databases on Different Nodes	133
104.	DUOW Configuration with Remote DB2/2 and Remote DB2/6000 Databases on Different Nodes	134
105.	DUOW Configuration with Remote DB2/2 and Remote DB2/MVS Databases Cataloged on the Same Node	135
106.	DBCS Client/Server Configuration Overview	142
107.	DBCS-Enabled Sample Application GUI Window	144
108.	STAFF Table Definition for DBCS-Enabled Environment	144
109.	Create Statement for DBCS-Enabled STAFF Table	145
110.	VisualAge Generator MVS CICS Generation Options	147
111.	CM/2 NDF File for Server Workstation Host Connection	150
112.	CICS Client Initialization File	153
113.	CICS OS/2 SIT Table Definitions	154
114.	CICS OS/2 Terminal Control Table	155
115.	Processing Program Table for CICS OS/2	157
116.	MVS CICS Connection Definition for CICS OS/2 System	158
117.	MVS CICS Session Definition for CICS OS/2 System	159
118.	MVS CICS Mirror Transaction Definition	161
119.	MVS CICS Program Definition	162
120.	Primary Window of VGPING Sample Application Client	166
121.	Server Data Window for VGPING Sample Application Client	167
122.	Structure of Tier-2 Server Application	167
123.	Primary Window for DUOW Sample Application Client	170
124.	Structure of DUOW Sample Application Server	170

Tables

1.	Summary of VisualAge Generator Client/Server Communication Support for Open Blueprint Integration Objectives	16
2.	DCE-Based Client/Server Communication Protocol Options	27
3.	Primary Software Installed on CICS OS/2 Server Workstation	30
4.	DCE-Based Client/Server Communication Protocol Options	54
5.	Common Problems Encountered During DCE Setup and Configuration	98
6.	Common Problems Encountered During DCE Runtime	100
7.	Common Problems Encountered During DUOW Testing	136
8.	Sample application Servers	165

Preface

This redbook examines VisualAge Generator client/server communications support and reviews selected run-time configurations. We focus on client/server configurations that use the CICS and DCE middleware options supported by VisualAge Generator and the use of DB2 distributed unit-of-work support in VisualAge Generator applications. A discussion of the role of middleware in an application system, the methods of implementing security and logical unit-of-work management, and the use of VisualAge Generator in a DBCS environment are included.

This redbook was written for application developers and service professionals involved in the design and implementation of VisualAge Generator client/server systems. Some knowledge of VisualAge Generator, CICS, and client/server design issues is assumed.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world, working at the International Technical Support Organization, San Jose Center.

Patrick F. McCarthy is a Consulting Application Development Specialist at the International Technical Support Organization, San Jose Center. He writes extensively and teaches IBM classes worldwide on all areas of application development with a specific emphasis on VisualAge Generator and related technologies. Before joining the ITSO in 1990, Pat worked in an IBM internal information systems organization (Corporate Component Procurement I/S) in Poughkeepsie, New York as a programmer, database administrator, and development center leader. Pat received a B.S. in Business Administration from the Indiana University of Pennsylvania (in Indiana, Pennsylvania) and an M.S. in Computer Science from Marist College.

Meltem Civelek is a Technical Marketing Specialist in the Software Marketing organization of IBM in Turkey. She has worked at IBM for 7 years and has 2 years of experience in application development projects using VisualAge Generator and VisualAge Smalltalk. Meltem assists customers in implementing VisualAge Generator three-tier applications using CICS, DB2, and Message Queuing technology. She has written extensively on Distributed Unit of Work implementation, Windows 95, and Windows NT configurations. She has a degree in Control and Computer Engineering from the Technical University of Istanbul.

Claus Diekmann is an Application Development Architect in Germany. He has worked at IBM for 11 years and has 5 years of experience in the area of application development.

Rich Gast is a World Wide VisualAge Generator technical sales specialist. He has worked at IBM for 13 years and has 8 years of experience in the field of client/server computing. His areas of expertise include product development in DB2, DFSMS, and ADSM in the areas of performance, client/server, and system management. He has written extensively on client/server, performance, and system management topics for both DB2 and ADSM.

Carlo Haverkamp is working as an Information Technology Specialist in the Professional Services organization of IBM Uithoorn in The Netherlands. He studied at the Vrije Universiteit Amsterdam and has a degree in Computer Science. During his study, he did research for IBM Uithoorn on CASE tools. Since joining IBM in September 1995, he has supported customers in installation and system development using VisualAge Generator products. He has technical expertise in various programming languages (Smalltalk, C++, JAVA, REXX), operating systems (AIX, UNIX, OS/2), and environments (CICS, DCE) and also has a wide theoretical background on many aspects of information technology. His main contributions to this book are the client/server configurations for AIX, CICS/6000 and DCE.

Fabrizio Lorenzini is a member of the Application Development Competence Center in Italy offering second-level field support for VisualAge Generator and Cross System Product. He has 5 years of experience in application development using multiple fourth generation language products. He has worked at IBM for 7 years with a focus on application development and client/server architecture.

Shengli Liu is a Technical Sales Specialist in the Peoples Republic of China. He has 5 years of experience in application development and is an IBM-certified DB2 Administrator.

Vitor Ventura is an Information Technology Specialist in Portugal. He has 11 years of experience in software support. His areas of expertise include application development, VM/ESA, and UNIX.

Joan Zhao is a technical marketing specialist in the Peoples Republic of China. She has 2 years of experience in application development, and has worked at IBM for 3 years. Her areas of expertise include VisualAge Generator client/server communications. She has written extensively on DBCS-enabled client/server configurations.

Thanks are also due to many people for their invaluable contributions to this project. From the VisualAge Generator development lab:

John Snyder **Kristine Heaton**

Roger Newton **ShuChen Shieh**

Deborah Foreman **Paul Hoffman**

VisualAge Generator supporters from IBM in the United Kingdom:

John Ormerod

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 185 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Home Pages at the following URLs:

For Internet users

<http://www.redbooks.ibm.com>

For IBM Intranet users

<http://w3.itso.ibm.com/redbooks>

- Send us a note at the following address:

redbook@vnet.ibm.com

Chapter 1. Introduction

VisualAge Generator provides multiple options for client/server communication support (see Figure 1) as part of the VisualAge Generator runtime environment support products (VisualAge Generator GUI Runtime Support, VisualAge Generator Workgroup Services, and VisualAge Generator Host Services).

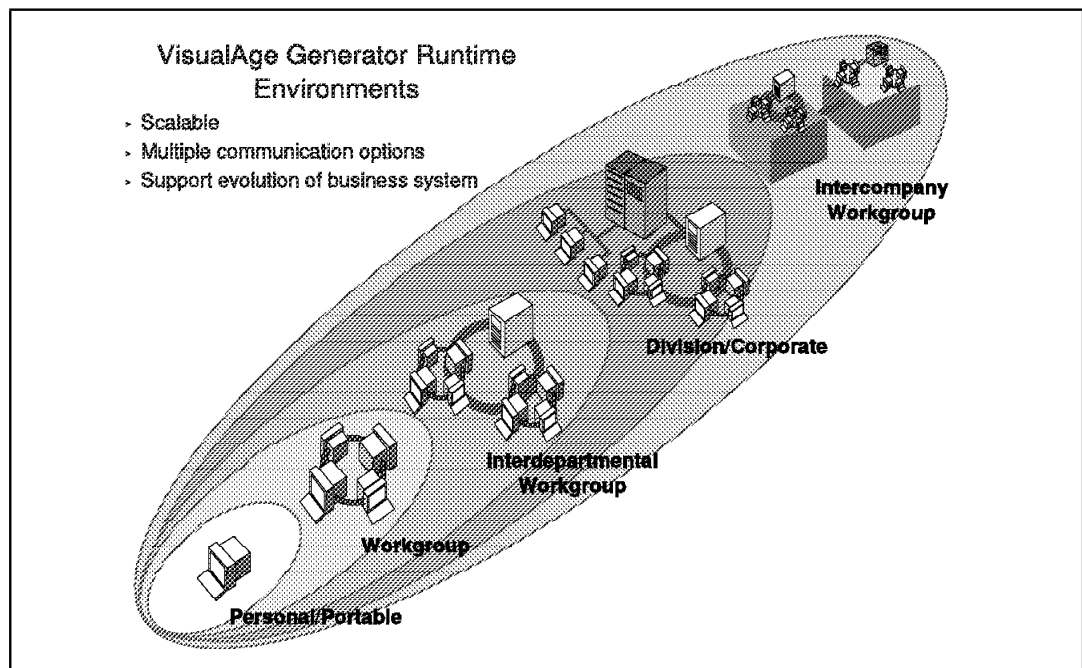


Figure 1. VisualAge Generator Client/Server Implementation Options

This book studies the issues associated with selecting and implementing one of the client/server communication configurations available for a VisualAge Generator client/server application system.

The residency project that produced this book focused on the implementation of selected client/server communication configurations so that we could explore the basic functions of VisualAge Generator client/server communication support and offer sample working configurations to help you in your environment.

Although the project scope was restricted to selected client/server communication configurations, the basic concepts behind the implementation of client/server communication can be understood and applied to any supported VisualAge Generator client/server target runtime environment.

In this chapter, we discuss:

- The focus of the project that produced this publication
- The role and responsibilities of each component in an open client/server system based on the Open Blueprint

- How each of the VisualAge Generator options for client/server communication provides support for the services defined in the Open Blueprint.

Before you read this chapter, you may wish to review "Chapter 1. Introducing Client/Server Application Development" in the *Developing VisualAge Generator Client/Server Applications* manual. The referenced discussion is short, but very appropriate; we therefore request that you at least review that chapter before reading our publication.

1.1 Project Scope and Objectives

VisualAge Generator provides a powerful development platform for building client/server application systems.¹ The capability provided by VisualAge Generator client/server communication enables you to connect VisualAge Generator clients and servers in an almost endless number of configurations that support a variety of communication services and protocols (see Figure 2).

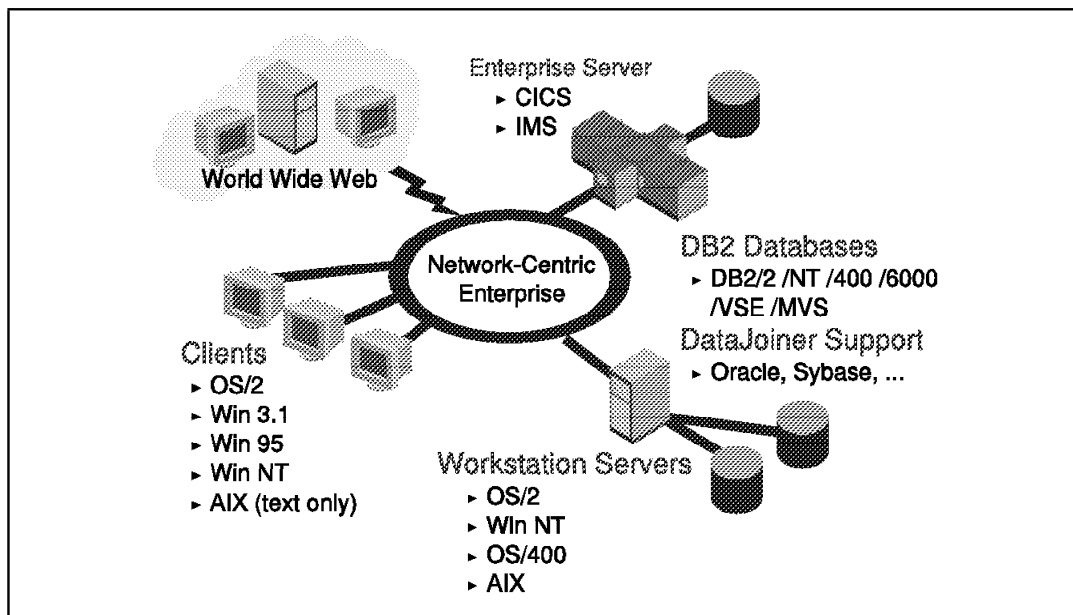


Figure 2. VisualAge Generator Client/Server Configuration Options

Even with the power of VisualAge Generator, the design and implementation of a client/server system presents a challenge. There are significant design decisions and implementation issues that must be resolved.

During implementation, there are additional issues that must be resolved. Our view of implementation issues is linked to the use of VisualAge Generator during development and the configuration options for client/server communication provided by and supported with VisualAge Generator.

In this publication, we focus on these implementation issues:

- How VisualAge Generator Developer can be used to improve the testing of client/server application systems.

¹ VisualAge Generator is also a very good tool for host (3270 screen) application development, but the focus of this book is client/server system implementation.

- What the major differences are between the multiple client/server communication and configurations options supported by VisualAge Generator applications in three areas:
 - Performance
 - Administration
 - Cost.
- How to configure a working VisualAge Generator client/server system using CICS- or DCE-based communication services options.

The work done to gain the experiences that are presented in this book reflect the functions available in VisualAge Generator Version 2.2.

We do not focus on the use of VisualAge Generator's proprietary middleware in this book. This option for client/server communication has already been studied and is documented in *Implementing VisualGen Client/Server Communication*, GG24-4235.

Our assessment of VisualAge Generator's client/server communication options was based on a view of open client/server systems as defined in IBM's Open Blueprint. This assessment shows that the client/server middleware options such as DCE and CICS which are not VisualAge Generator proprietary, are the more appropriate and strategic choices.

Given our focus and belief in the value of these open client/server communication options we discuss arguments for a transition from the VisualAge Generator proprietary middleware to these other client/server communication options.

This book does not attempt to explain all of the functions, options, and configuration issues related to implementing a VisualAge Generator client/server application. It is not intended to be your only guide in the process of implementing a VisualAge Generator client/server application system. You should also use the publication *Developing VisualAge Generator Client/Server Applications* as a reference and source for initial guidance.

Significant skill with the underlying technologies used to support VisualAge Generator client/server communication may be required to configure a working application system.

1.2 Positioning VisualAge Generator Client/Server in the Open Blueprint

In this section, we examine how VisualAge Generator and supported communication services components, such as Distributed Database Architecture (DRDA), Customer Information Control System (CICS), Distributed Computing Environment (DCE), and VisualAge Generator proprietary middleware support fit within the open client/server architecture of the IBM Open Blueprint.

The IBM Open Blueprint provides an architecture for open and flexible client/server solutions. The IBM Open Blueprint is discussed in detail in:

- *Introduction to the Open Blueprint: A Guide to Distributed Computing*
- *Open Blueprint Technical Overview*

In the Open Blueprint, the processes and functions of an open client/server system are positioned in an architecture of interrelated application system components (see Figure 3 on page 4).

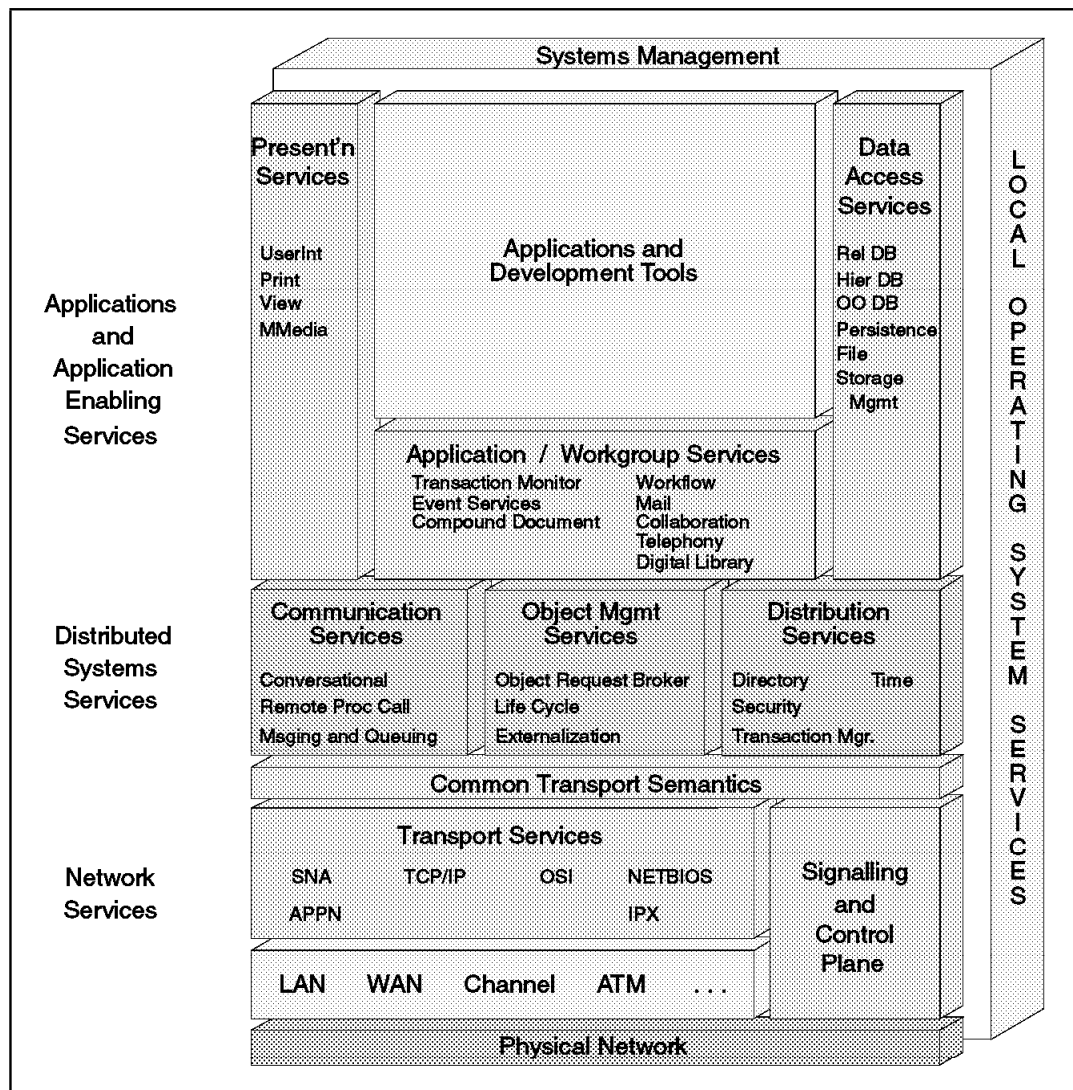


Figure 3. Components of the Open Blueprint

1.2.1 Open Blueprint Components

The components of the Open Blueprint are listed below:²

Application-enabling services:

Presentation services define the interaction between applications and the user.

Applications and development tools help the application developer implement distributed applications that use standard interfaces and the facilities of the Open Blueprint.

Application/Workgroup services provide common functions, such as mail, which are available for use by all applications.

² The Open Blueprint descriptive information presented in this chapter is from *Introduction to the Open Blueprint: A Guide to Distributed Computing*.

Data access services allow applications and resource managers to interact with various types of data.

Distributed-systems services:

Communication services provide mechanisms for parts of a distributed application or resource manager to talk to each other.

Object management services provide common object services including transparent access to local and remote objects.

assist the communication between parts of distributed applications and resource managers by providing common functions such as directory and security.

Network services:

Common transport semantics support protocol-independent communication in distributed networks. Transport services provide the protocols for transporting information from one system to another, such as SNA/APPN*, TCP/IP, OSI, NETBIOS, and IPX**.

Signalling and control plane provides the ability to establish subnetwork-specific connections.

Subnetworking provides functions dealing with specific transmission facilities, such as various kinds of LANs, WANs, channels, asynchronous transfer mode (ATM) and emerging technologies such as wireless.

Systems management:

Provides facilities for a system administrator or automated procedures to manage the network operating system.

Local Operating System Services:

Operate within the confines of single system in a network. Examples of local services are managing memory and dispatching work.

The components defined as part of the Open Blueprint can help you to understand the role and responsibilities that each logical component of an open client/server system should perform. One or more products, or your application system if you choose to write your own component services type of function, should implement the services that are specific to each component in the Open Blueprint. In a client/server system, some of these services are critical, so we review them in detail.

1.2.2 Critical Services

When considering the implementation issues related to building client/server systems we focus on these services that are provided by components of the Open Blueprint:

Application/Workgroup services

Application services provide high-level application- or workgroup-oriented functions. They include:

Transaction Monitor

Transaction monitor is an industry term for functions that traditionally have been included in IBM's transaction processing systems. The Transaction Monitor provides an environment for the development and execution of applications, embodied in the transaction programs. The monitor typically provides an application programming interface and support for efficient transaction execution.

The transaction monitor supports a large number of users concurrently sharing access to the transaction programs and the resources they use. Transaction monitors allocate system and application resources

beforehand, such as address spaces, data, and other facilities. The preallocation allows transaction programs to be scheduled efficiently.

The transaction monitor uses the transaction manager directly, in many cases, to simplify the application programming implementation of the transaction. A transaction monitor also typically provides some application development and system management support for the transaction programs. No formal standards have yet been established for the transaction monitor application programming interfaces.

The CICS transaction monitor API has been implemented on all major IBM platforms and on many non-IBM platforms. The IMS transaction monitor API has been implemented on a variety of platforms supporting applications associated with MVS systems. The Encina transaction monitor API has been implemented across a range of UNIX platforms.

Distribution services

Distribution services make a single-system view of the network possible.

Naming and directory

The naming and directory service provides a consistent approach to naming and keeping track of network resources and their attributes.

Naming provides the facilities required to refer to such network resources as servers, files, disks, applications, and disk queues. The use of a consistent naming model allows a resource to be accessed by name, even if a characteristic such as its location is changed.

The directory service maintains information about the characteristics of a resource, such as its name, network address, and creation date.

Security

The security service protects network resources from unauthorized use by registering users (both system and human) and their authorization levels, by authenticating users, and by auditing access.

In earlier centralized systems, the operating system authenticated user identities and authorized access to resources. Individual workstations in a network are not necessarily secure. Therefore, in a distributed environment, security operations must be performed by an independent set of services. Security in a distributed environment must support single sign-on and address such challenges as preventing eavesdropping, impersonation, and forgery.

The Open Blueprint security service specification lets administrators register users and resource managers, provides for the mutual authentication of clients and servers, and enables resource managers to provide access to resources only to authorized users. The Open Blueprint security service also defines services for auditing user activity. These services include DCE specifications and incorporate and expand on the Kerberos specification from MIT. These security services meet relevant X/Open and POSIX specifications.

Time

The time service regulates the date and time across a network. The transaction manager coordinates resource recovery across the various systems in the network.

The systems and applications that operate across a distributed environment need a consistent time reference to schedule activities (such as recovery) and determine the sequence and duration of events. Different components of a distributed application obtain time from clocks on different computers. The distributed time service, based upon DCE, synchronizes system clocks in a network to provide time services with a limited, but known degree of accuracy for distributed applications.

Transaction manager

In transaction processing, application processing is divided into units of work called *transactions*. A transaction may involve only a limited number of interactions with a user at a workstation, but it may involve many interactions with many resource managers across the network. The transaction manager provides synchronization services so that multiple resource managers can act together to ensure that resources retain their integrity. The resources managed separately by each remain consistent according to relationships imposed externally, typically by the application.

The current use of the term *transaction manager* differs from earlier usage. This new terminology has been adopted to accurately reflect technical goals and the functional parts of the Open Blueprint and to correlate with standard industry terminology.

Major IBM products such as Customer Information Control System (CICS), Encina**, and Information Management System (IMS*) are combinations of the transaction manager, the transaction monitor, and other functions.

A distinguishing feature of transaction processing is that all the resource changes associated with a transaction must be committed before the transaction is complete. If there is a failure during execution of the transaction, all of the resource changes must be removed. Resources managed in this manner are recoverable.

These services should be implemented as part of a client/server system. If not, their absence should be acknowledged so that application design changes can be considered, if required. This is not the same as saying that these services should be implemented by VisualAge Generator.

The main message we take from the IBM Open Blueprint in the context of this book about VisualAge Generator client/server communications implementation is that each component performs a specialized task and is both providing services to, and using services of, other components defined in the architecture.

Thus VisualAge Generator (a development tool) and the generated VisualAge Generator applications do not deliver these types of services themselves but rely on other components in the system that are specifically designed to perform these other services.

What we must assess is the combination of VisualAge Generator applications in an established client/server environment based on one of the available VisualAge Generator client/server communication options. How well the combined technologies and components of the Open Blueprint are integrated determines the overall value of the solution.

1.2.3 Integration Objectives

The guiding integration objectives that should be considered when making client/server communication decisions are also identified in the Open Blueprint documentation. The following scenarios are key focus areas for integration:

Single Sign-on

Lets the user have a single identification within the network. Here, *network* could refer to one business or physical network or to multiple networks. Single sign-on lets users log on with a single password and have access to all the network facilities for which they are authorized.

Network-wide security

Protects network resources and users. It encompasses three basic areas:

- Data encryption to protect data in the network
- Authentication of users and resource managers

- Resource access control to manage what a particular user can do

Authentication involves identifying the client and validating the server.

Network-wide directory

Provides information about resources in a network.

The directory eliminates the need for product-unique ways to locate resources, and shields users from keeping track of where resources are located.

Global transparent access

Enables a user to access data or applications in a network or networks without concern for where they reside.

Each of the client/server communication options supported by VisualAge Generator supports the basic services and integration objectives defined in the Open Blueprint as discussed in the next section.

1.3 VisualAge Generator Communication Services Capabilities

VisualAge Generator provides several methods of implementing client/server or distributed computing (see Figure 4).

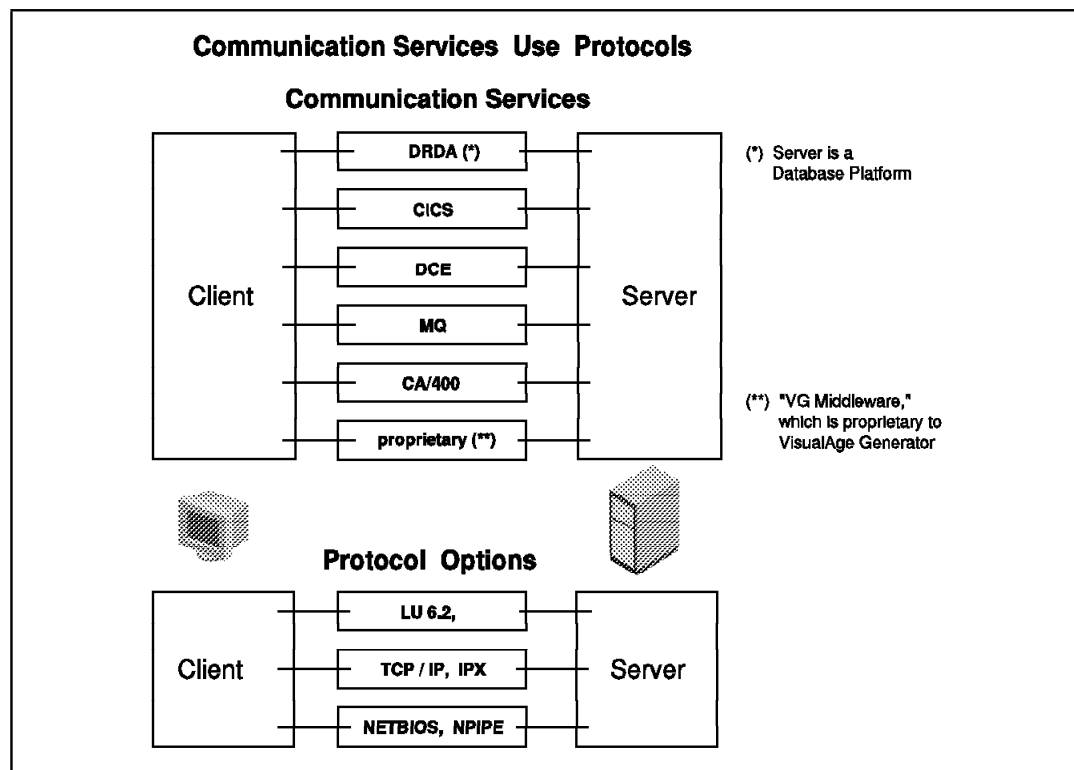


Figure 4. VisualAge Generator Client/Server and Distributed Computing Options

VisualAge Generator's primary support for client/server systems is based on the use of a remote procedure call (RPC) communication service between the client and the server. This allows programmers to simply use the CALL statement to call server applications and not worry about how the CALL is implemented at run time. VisualAge Generator and client/server communication implement the difficult parts of cross-system client/server RPC support.

VisualAge Generator applications can also use the conversational and message queuing flavors of a communication service, but this support requires more design

and coding work on the part of the application programmer. These options are not discussed in this book.

VisualAge Generator client/server configurations can also be based on the use of remote database access as provided by DRDA. Although DRDA is not used directly by VisualAge Generator applications but by the Database Management System (DBMS) on behalf of VisualAge Generator applications, we mention it here because from the view of a system architect, DRDA is another valid choice when designing a client/server application system. The DRDA configuration option is discussed in Chapter 5, "Database-Enabled Client/Server" on page 103.

Several methods are available for the implementation of RPC support when using VisualAge Generator:

CICS-based client/server communication

Supports calls to VisualAge Generator servers in CICS runtime environments.

DCE RPC support (secure and nonsecure)

Supports calls to VisualAge Generator servers on Windows NT, OS/2, AIX, and MVS runtime environments.³

VisualAge Generator middleware client/server communication

Supports calls from workstation platforms to other selected workstation or host runtime environments.⁴

In 1.2, "Positioning VisualAge Generator Client/Server in the Open Blueprint" on page 3, we identified services that should be provided as part of a client/server system. We also identified integration objectives that can be used to guide client/server communication configuration decisions. In the following sections we review the support for these services and integration objectives as provided by the available VisualAge Generator RPC implementation options. We also consider the issues identified in 1.1, "Project Scope and Objectives" on page 2.

1.3.1 CICS

VisualAge Generator has an affinity for CICS. Many VisualAge Generator client/server and 3270 stand-alone application functions are optimized for the CICS environment. VisualAge Generator applications can be implemented on almost all of the available flavors of CICS (MVS, VSE, OS/2, and Windows NT).

Our review of VisualAge Generator client/server systems that use the CICS-based implementation of RPC support assumes the use of CICS Client or CICS OS/2 Client software for client/server communication support. The use of a mixed environment where VisualAge Generator middleware is used to connect to CICS is discussed in 1.3.3, "VisualAge Generator Middleware" on page 13.

³ VisualAge Generator intends to provide support for DCE RPC calls to MVS servers in future updates to VisualAge Generator client/server communication support.

⁴ The VisualAge Generator middleware support provided with V1.0 was based on technology licensed from another company. This licensed middleware function, packaged as part of VisualAge Generator runtime support products, continues to be available in V2.2 of VisualAge Generator. IBM developed VisualAge Generator middleware options for implementing VisualAge Generator client/server communication RPC support, such as IMS APPC and CA/400, which are also available with V2.2.

CICS support for the Open Blueprint services, integration objectives, and implementation issues identified previously are reviewed below:⁵

Transaction Monitor	Excellent. CICS, both a transaction monitor and a transaction manager, is used by thousands of customers around the world for mission-critical application systems. The CICS transaction monitor API has been enabled on all major IBM and many non-IBM platforms.
Naming and Directory	Excellent. CICS naming conventions and the ability to refer to names in one CICS region that are physically implemented in another connected CICS region provides application design flexibility and support for distributing the workload required to manage different resources. (Think in terms of file-owning regions, application-owning regions, and terminal-owning regions.)
Time	Strong in a CICS-based network on a set of hosts or on a set of workstations. Each CICS region will present time and date based on the local operating system. CICS-based networks do not coordinate time but there are methods of achieving this at the operating system level.
Transaction Manager	Excellent on MVS, VSE, AIX, and Windows NT platforms. Strong on OS/2 and Windows NT platforms. ⁶ CICS is both a transaction manager and a transaction monitor. CICS fully implements the required support for recoverable resources across multiple platforms when a server-based LUW management design is implemented. Not all CICS platforms can provide complete resource management for a client-based LUW design. Because of this and, more important, to increase platform portability of your application system and overall systems performance, a server-based LUW management approach is recommended regardless of the option chosen for client/server communications.

⁵ We have combined the assessment of the Open Blueprint security service support with the single sign-on and network-wide security integration objectives. This discussion point is termed *system security*.

⁶ CICS OS/2 provides full support for recoverable resources when the resource is owned by CICS (files as well as recoverable transient data queues and temporary storage). When other resources are used, such as a relational database, CICS OS/2 provides a level of resource commitment coordination but does not fully implement a coordinated resource management environment. (For example, a CICS synch point is processed independently, but in concert with, a DB2/2 commit work.) This means that CICS OS/2 does not implement complete resource management support for relational databases (such as DB2/2) while CICS/6000 does provide complete resource management support for IBM and non-IBM relational databases.

The currently available version of CICS NT is based on the CICS OS/2 product. The next version of CICS NT will be based on the CICS/6000 product. The resource coordination support for CICS NT will equal that provided by the CICS/6000 code used as a base.

VisualAge Generator significantly reduces concern over this issue when a server-based logical unit of work (LUW) management design is used. VisualAge Generator will expand commit and rollback requests as follows:

EZECOMIT	CICS SYNCPOINT and SQL COMMIT WORK
EZEROLLB	CICS SYNCPOINT CANCEL and SQL ROLLBACK

⁷ Depends on the CICS release level. Recent versions of CICS for host platforms support other security managers only; that is, CICS stopped providing its own security system.

System Security	Excellent in a single system, Strong in a CICS-based network. CICS has its own security system as well as support for other security managers.⁷ These other security managers can be IBM products such as RACF on the host or UPM on the workstation. Non-IBM products on the host include options such as CA-ACF2 and CA-TOP SECRET. VisualAge Generator user authentication support provides options on how the user ID and password are obtained from the end-user or end-user's workstation configuration. A fairly secure option is the VisualAge Generator-provided user exit that uses a dialog to obtain the user ID and password from the end-user that will be passed to the selected client/server communications option. End-user logons to a LAN server resource are not automatically detected and used for client/server communication user authentication and authorization. Doing so would require customized code. CICS propagates user ID and password details from a CICS Client entry point to the target CICS region and throughout a CICS-based network. CICS OS/2 provides support for using the workstation-based security manager User Profile Manager (UPM) as the authentication and authorization check point. Other configurations allow CICS OS/2 to bypass security checking and let other connected CICS regions implement authentication and authorization processing.
Network-wide directory	Strong. CICS can support directory processing by chaining CICS regions together. Definitions in one region can point to other regions where the resource may exist (or be redefined to another location).
Global transparent access	Strong. CICS-based servers can be on either local or remote platforms. The operating system used to support the CICS target environment is not a major concern when using the service. VisualAge Generator applications can communicate across CICS region boundaries. ASCII/EBCDIC conversion issues can be fully automated by VisualAge Generator and ignored by the application programmer in most situations. When required, VisualAge Generator application programmers can take control of ASCII/EBCDIC conversion processing to implement special requirements.
Performance	Excellent. The fastest configuration available for VisualAge Generator client/server system using a CICS target server platform uses a CICS-based client/server communication configuration. That is, connect your clients to CICS servers using CICS Client software if at all possible! DCE- and VisualAge Generator middleware-based client/server communication options support CICS target server platforms, but each performs a bit slower than a CICS Client-enabled configuration. (The performance implications of DCE- or VisualAge Generator middleware-based client/server communication are reviewed under those headings.) Actual performance depends on network load, message size, target server platform, server workload, and client/server communication configuration.

Administration	Strong. Defining a connection between a CICS Client-enabled workstation and a target CICS OS/2 or CICS NT workstation requires that one file be updated (see 3.2.3, "CICS Client" on page 44). CICS Client provides support for the NetBIOS, TCP/IP, and IPX protocols when connecting to workstation targets. MVS and VSE connections require SNA LU6.2 definitions and sessions.
Cost	A Factor. While CICS Client software is an addition to the requirements for an end-user client workstation, we feel the benefits outweigh the issue of the additional configuration requirement in the long term.

1.3.2 Distributed Computing Environment

Support for DCE-based client/server communication is new with VisualAge Generator V2.2. A choice of DCE for client/server communication support may be partially based on its merits (as assessed below) and partially on its strategic nature. DCE is viewed as an open, industry-supported, standard method of implementing secure client/server systems.

Organizations may decide to put significant emphasis on DCE as part of a network application-support strategy. By providing a DCE-based client/server communication option, VisualAge Generator is moving toward a role as a strategic open client/server system enablement tool.

Transaction monitor	None when DCE is used to support client/server communication with servers on native OS/2 or AIX platforms. Excellent when DCE is used to support client/server communication with servers on IMS/DC and CICS MVS platforms. ⁰⁰ VisualAge Generator applications do not obtain transaction manager services from the DCE environment. Only when the application is running in a CICS or IMS/DC runtime environment do transaction management services exist.
Naming and directory	Excellent. The fundamental architecture of DCE enables naming and directory services.
Time	Excellent. While applications running on different platforms obtain time information from the local operating system, DCE configurations enforce the synchronization of time on multiple connected workstation operating systems.
Transaction manager	None when DCE is used to support client/server communication with servers on native OS/2 or AIX platforms. Excellent when DCE is used to support client/server communication with servers on transaction processing platforms such as IMS/DC and CICS MVS. VisualAge Generator applications do not obtain transaction manager services from the DCE environment. Only when the application is running on a transaction processing platforms such as CICS or IMS/DC do transaction management services exist.

System security	Excellent. The method of managing user ID and password data in a network used by DCE is very secure. VisualAge Generator supports two DCE-based client/server communication options: DCE and DCESECURE. The secure DCE option increases the level of control by ensuring that the DCE configuration allows a specific user to access a specific server (DCE service). A DCE cell provides a single point of user definition, authentication, and authorization control. DCE will propagate user ID and password details from a DCE entry point to the target server platforms. This would allow one user ID/password pair to be used for native OS/2 and AIX server access and also the MVS CICS or IMS/DC host server platforms also supported by DCE.
Network-wide directory	Excellent. A DCE cell provides a single point of server (service) definition and the mapping to the physical location where the server (service) exists.
Global Transparent Access	Excellent. All requests for a server (service) that is accessed using DCE-based client/server communication support are resolved at a single point. This protects users from having to know if or when changes take place in how servers (services) have been distributed in the network.
Performance	Excellent. While there is some overhead in a DCE-based client/server communication environment because of the directory lookup associated with the first CALL to a server on a given target platform, subsequent calls know where the servers are located and no lookup is required. (See 4.2, "DCE Client/Server Scenarios" on page 56 for details.) Actual performance will depend on network load, message size, target server platform, server workload, and client/server communication configuration.
Administration	Strong. A DCE-based client/server communication environment can be configured from a single location.
Cost	A Factor. While DCE Client software is an addition to the requirements for an end-user client workstation, we feel the benefits outweigh the issue of the additional configuration requirement in the long term. A DCE cell (directory) configuration is also required. This software is typically more expensive. However, a DCE cell can be configured on one of many different operating system platforms and can be used across a network of different workstation types.

1.3.3 VisualAge Generator Middleware

VisualAge Generator has provided a proprietary middleware solution for client/server communication since the first release of the product. The other options (pure CICS- and DCE-based client/server communication) have been added since the first release.

The VisualAge Generator middleware option is initially attractive because it is included as part of the VisualAge Generator runtime installation. But the VisualAge Generator middleware option also has a reputation for being difficult to configure, slow, and not fully reliable in some situations. (You could say you get what you pay for in this situation.)

Many configurations are supported by VisualAge Generator middleware functions. (These are reviewed in detail in *Implementing VisualAge Generator Client/Server Communication*, GG24-4235.) Some of the available options, such as LU2 support, are extremely valuable when a client workstation does not have a LAN connection (token ring or Ethernet) to the enterprise network.

Other options, such as using TCP/IP to connect clients with servers running on OS/2 and AIX are attractive because no additional software is required, keeping costs down, but the configuration lacks support for the services and integration objectives identified earlier.⁸

Using VisualAge Generator middleware to connect with CICS targets, given that a gateway configuration that requires two hops to satisfy the server request is mandatory, is not viewed as a reasonable option when compared with the performance and relative ease of configuration of a CICS-based client/server communication solution.

Transaction monitor	None when used to connect with native OS/2 or AIX applications. Excellent when used to connect with IMS/DC.⁹ Average when used in a gateway (two-hop) configuration with CICS as the target server platform.
Naming and directory	Average. VisualAge Generator middleware configurations can use either a linkage table or both a linkage table and an RTABLE as part of a client/server communication configuration to control the mapping of server request to physical location. Linkage tables are required with other client/server communication options but they can be one-line entries that basically say "go ask CICS" or "go ask DCE" where the server is located.
Time	Average. Applications running on different platforms obtain time information from the local operating system. Methods exist to synchronize time on multiple, connected, workstation operating systems. Designs that solicit time from a server on a trusted platform are recommended when time (date) information is critical.
Transaction manager	None when used to connect with native OS/2 or AIX applications. Excellent when used to connect with IMS/DC. Average when used in a gateway (two-hop) configuration with CICS as the target server platform.

⁸ The VisualAge Generator development lab may provide middleware support for TCP/IP client/server communication processing between selected client and server platforms in the future. Customer requests for this connection option are being evaluated. The services provided may not equal those provided by the CICS- or DCE-based client/server communication options.

⁹ VisualAge Generator VisualAge Generator middleware support for the IMS/DC target is not as closed as other VisualAge Generator middleware options. See 'Appendix E. VisualAge PowerServer APIs' in the *Developing VisualAge Generator Client/Server Applications* product manual.

System security	Excellent to Limited. Depends on target platform and configuration. VisualAge Generator middleware provides options for local exits to perform user authentication and authorization. Gateway functions will propagate the user ID and password provided at the client through to the next hop on the call to a server. Local control of user ID authentication and authorization is required. It is possible to use the exits to link VisualAge Generator middleware with other security managers such as UPM or to just pass the user ID and password to the target runtime platform for actual authorization processing.
Network-wide directory	Strong to Limited. Depends on target platform and configuration. The directory is implemented as part of the client/server communication configuration defined in the linkage table and, if used, the RTABLE.
Global transparent access	Strong to Limited. Depends on target platform and configuration. As with the directory, access is via the client/server communication configuration defined in the linkage table and, if used, the RTABLE.
Performance	Strong to Limited. Depends on target platform and configuration. When the target platform is CICS, there is a penalty for using the VisualAge Generator middleware client/server communication option. The mandatory gateway configuration when targeting a workstation CICS environment adds a second hop (call propagation) to the server request. Actual performance will depend on network load, message size, target server platform, server workload, and client/server communication configuration.
Administration	Limited. VisualAge Generator middleware configurations are more difficult. The use of a linkage table and one or more RTABLEs and DNA.INI control files makes the whole process of configuring a working client/server system more difficult.
Cost	A Factor. VisualAge Generator middleware support is provided as part of the runtime environment for VisualAge Generator applications. The only problem with using VisualAge Generator middleware is that for all configurations, except those with IMS/DC and OS/400 targets, the middleware cannot be reused by non-VisualAge Generator applications.

1.3.4 Summary

Table 1. Summary of VisualAge Generator Client/Server Communication Support for Open Blueprint Integration Objectives			
Objective	CICS	DCE	VisualAge Generator Middleware
Transaction monitor	Excellent	Excellent with MVS CICS or IMS/DC server targets. None when used with workstation server targets.	Strong with MVS CICS or IMS/DC server targets. None when used with workstation server targets.
Naming and directory	Excellent	Excellent	Basic
Time	Strong	Excellent	Strong
Transaction manager	Excellent	Excellent with MVS CICS or IMS/DC server targets. None when used with workstation server targets.	Strong with MVS CICS or IMS/DC server targets. None when used with workstation server targets.
System security	Excellent to Strong	Strong with MVS CICS or IMS/DC server targets. Excellent when used with workstation server targets.	Basic
Network-wide directory	Strong	Excellent	Basic
Global transparent access	Strong	Excellent	Basic
Performance	Excellent	Excellent	Strong to Limited
Administration	Excellent	Strong	Strong to Limited
Cost	A factor	A factor	Not a factor (provided by VisualAge Generator)

1.4 Implementing Client/Server Systems with VisualAge Generator

The process of selecting from the list of available protocols and configurations can be difficult. This book attempts to guide you through the complexities.

There are numerous implementation and configuration options for client/server communication, and you must understand these options and make decisions about them early in the development cycle. Selecting a client/server configuration requires that application designers or architects and VisualAge Generator programmers understand the functional and operational attributes of each possible configuration.

In making decisions about the design and implementation of a VisualAge Generator client/server application system, ask the following questions:

- What are the protocol options for VisualAge Generator client/server communication?
- Which option best fits my needs and current systems environment?
- Which LUW management options are available in each client/server communication configuration?

- If data conversion is required, how will it be managed?
- Can I change my client/server implementation choice without affecting the overall function of the VisualAge Generator client/server application system?
- Will testing VisualAge Generator client/server applications using the VisualAge Generator Developer Interactive Test Facility (ITF) provide a true simulation of the operation of the application in the target runtime environment?

In the rest of this book, we review the available VisualAge Generator client/server configuration options.

Chapter 2. Testing Applications with VisualAge Generator Developer ITF

One of the powerful aspects of VisualAge Generator is the fact that you can test the source code of your application independent of the runtime environment by using the ITF. Database identification and access is controlled by the ITF.

In an ITF environment, both GUI and server applications run in a fully interpreted mode; that is, testing using ITF simulates the runtime environment.¹⁰ Of course, because the entire application (clients and servers) are interpreted during testing in ITF, processing is slower than it would be during runtime.

The ITF can also be configured to call executable versions of called applications (or non-VisualAge Generator programs) instead of continuing to interpret, in test mode, the called application.¹¹

Using executable versions of called applications can affect database identification and access as well as LUW management.

VisualAge Generator Developer ITF processing is enhanced with V2.2 to provide support for calling generated applications while still running some code from the member specification library (MSL) while both the executable and the MSL versions access the database. Commit-point processing is coordinated between ITF and the runtime environment.

Please consult the appropriate VisualAge Generator documentation for additional details.

This chapter discusses how calling generated applications is implemented by the ITF and the implications for database identification and access authorization.

2.1 Calling Generated Applications

When a call statement is detected during ITF source code testing, the call to a called batch application (server) can be processed such that it is either:

1. Executed in interpretive mode using the server code in the MSL
2. Implemented with a call to a generated local or remote application

¹⁰ The **goal** of the ITF is to provide an *exact* simulation of the actual runtime environment. This goal is met in most situations. Differences can occur when there are inconsistencies in the client/server configuration, LUW management control techniques, and environment variable settings. If you think ITF is not simulating the runtime behavior of your application, use your product support process to tell the IBM VisualAge Generator development lab, which wants to know about and, if possible, correct the inconsistency.

¹¹ Search on the text string "Calling external programs" in the VisualAge Generator Developer help facility for more guidance on using linkage tables with the ITF.

The VisualAge Generator linkage table is used to tell the ITF to directly call generated server applications in the server runtime environment. The active linkage table, if one is defined in the ITF general preferences profile, is searched for a :CALLLINK entry that matches the called application (or non-VisualAge Generator program) name. The use of executable code will occur if a :CALLLINK linkage table entry is found for the application or program that is being called. If a match is not found in the linkage table referenced in the ITF general preferences profile, then testing will continue using the called application source found in the active MSL.

2.1.1 Runtime Processing of Remote Calls

When the ITF calls a generated application the processing is very similar to what happens when a generated GUI client application calls a server. When a generated GUI client application calls a remote server application:

- An accessible linkage table file indicates how the call should be implemented.
- The client/server communication configuration required to call the server applications should also be enabled.

The active linkage table at runtime is the first linkage table found of either:

1. The linkage table identified during generation, minus the path information (The linkage table is searched for in the current directory and then in the DPATH.)
2. The linkage table referenced by the active setting of the CSOLINKTBL environment variable (If path information is not provided, the current directory is searched. DPATH is not searched.)

2.1.2 ITF Processing of External Calls

Linkage table processing for an ITF call differs slightly from that used during runtime. ITF uses the existence of the application in the linkage table identified in the ITF general preferences profile to determine whether a call is to be implemented using interpreted source code or a generated application, and if a generated application, whether the call will be a local or a remote call. Two methods are defined as part of the LINKTYPE option in the matching linkage table entry for how the application (or program) is called in executable form by the ITF:

LINKTYPE=oslink

A local call to an executable running on the same platform as the VisualAge Generator Developer (OS/2) is issued. The called application or program is located using the LIBPATH settings for the operating system.

LINKTYPE=remote

A remote call that is resolved using VisualAge Generator client/server communication support. Standard client/server communication processing logic is used. This begins with a **second** review of the active linkage table. Client/server communication call processing logic rereads the active linkage table. At this stage the active linkage table is the first found of:

1. A linkage table with the same name as that referenced in the ITF general preferences profile (The drive and directory information is not used to locate the linkage table. Client/server communication processing searches for the linkage table in the current directory and then in the OS/2 DPATH configuration setting.)
2. The linkage table referenced by the active setting of the CSOLINKTBL environment variable (If path information is not provided, the current directory is searched. DPATH is not searched.)

Note: This processing neither guarantees nor requires that the linkage table identified in the ITF general preferences profile is that referenced by the CSOLINKTBL environment setting.

2.2 Database Identification and Authorization

The methods and options for database identification and authorization processing are affected by several factors:

- Use of ITF interpreted logic
- Database environment variables
- Use of calls to generated applications
- Client/server communication environment variables

Database identification controls the name of the database to be used when SQL statements are issued in an application running in a workstation environment.

Database authorization determines whether the application and the user are allowed to access the tables in the target database and what qualifier is to be used for unqualified table names.

2.2.1 ITF Database Identification

The ITF uses the VisualAge Generator Developer database profile settings to identify the active database and other relevant database control information.

When the application is run from the MSL, any database access is based on the DB2/2 configuration on the developer's workstation. Access might be restricted to local DB2/2 databases, or, by using distributed database support, to a database on another workstation or host platform.

To identify a database to be used during the test session when no explicit database connects have been coded in the application, indicate the database name in the VisualAge Generator Developer profile. Select **Profile** and then **Database preferences...** to change the name of the database used.

To issue SQL statements the VisualAge Generator Developer must be bound to the target database. The first time the database is accessed through ITF or other VisualAge Generator Developer SQL activity (such as SQL Record definition), VisualAge Generator binds the eze2db2.bnd package to the database.

In addition to allowing VisualAge Generator Developer and the ITF to interact with the database, the bind also determines the format in which date and time values are returned to VisualAge Generator. If the format is incorrect, you can manually bind the package to the database using the correct date/time format parameter:

```
db2 bind eze2db2.bnd datetime XXX
```

The XXX indicates the datetime format to be used. See the DB2 documentation appropriate for your DB2 database system for more information about binding a package to the database.

2.2.2 ITF Database Authorization

When ITF interprets application logic all SQL activity is dynamic. If database tables are not explicitly qualified, the user profile manager (UPM) node logon ID (for a remote database) or the UPM local logon ID is used to determine access authority and as the table qualifier for unqualified tables. The UPM node logon ID defaults to local logon ID if a node logon was not explicitly performed. See 5.2.1.3, “Identifying the Database Authorization ID” on page 107 for a detailed discussion of UPM processing.

If you are going to access a remote DB2/MVS database, you can identify the ID to be used to qualify table names in the VisualAge Generator Developer database preferences profile.

The use of the CSOUEXIT environment variable (see 2.2.4, “Runtime Database Authorization” on page 23) can override the UPM value if the ITF is used to call generated applications. When a generated application is called, the runtime configuration is used to control processing. If the CSOUEXIT environment variable is set for runtime processing, this can impact subsequent VisualAge Generator Developer and ITF database authorization. The user ID and password obtained from the CSOUEXIT identified authentication routine will be used for all subsequent SQL processing by VisualAge Generator Developer and the ITF.¹²

CSOUEXIT authentication is used for both database access and remote call security (such as a call using the CICS Client ECI) so you may have to coordinate user ID/password values across multiple platforms when you mix ITF SQL access and calls to local and remote generated applications.

2.2.3 Runtime Database Identification

If you call a generated version of an application from the ITF, you may use a different database than you would if the MSL source was interpreted. This is because the database used by the generated application is identified by an environment variable while ITF interpreted logic accesses the database identified in the VisualAge Generator Developer database preferences profile (which defaults to the EZERSQLDB environment variable if a value is not provided).

When ITF calls a generated application, database access is based on the physical location (workstation or host) of the called application (or program) and the database configured for the target runtime platform.

The database that is used during runtime execution of the application on a workstation is determined by one of the following environment variables (when no explicit database connects have been coded in the application):

- FCWDBNAME_appl
- ELARTRDB_tttt
- EZERSQLDB
- DB2DBDFT

See 5.2.1.4, “Identifying the DB2 Database” on page 108 for a detailed discussion on the use of these environment variables. The setting of the appropriate environment variable can be changed before you start the application or call the application from the ITF. You may need to start the VisualAge Generator Developer environment from an OS/2 command line if you want to customize any database or client/server communication environment variable settings.

¹² Note that this processing may change if Fixpaks are applied to VisualAge Generator.

Note: Environment variable changes made in a command session (OS/2 window) are local. Global settings must be made in the CONFIG.SYS file.

Other environment variables can affect database selection for VisualAge Generator applications. Please review *Running VisualAge Generator Applications on OS/2, AIX, and Windows* for additional guidance.

2.2.4 Runtime Database Authorization

Database authorization processing is impacted by the client/server communication configuration and target runtime environment.

Local calls can use UPM or CSOUEXIT processing to obtain the authorization ID used for database access.

Remote calls to CICS targets will use CSOUEXIT processing to obtain the user ID and password used for the CICS ECI interface. The user ID identified in the ECI call is not always used for database authorization. Processing depends on the target CICS platform and configuration.

If you have used the CSOUEXIT environment variable to configure client/server communication control for user authentication, you can impact the database authorization ID used by ITF after a local or remote generated application has been called.

Chapter 3. CICS-Based Client/Server

VisualAge Generator provides very good support for CICS-based client/server communication. VisualAge Generator server applications can be generated for most of the available CICS platforms. Multiple approaches (two-, three-, and N-tier) to client/server system configuration are available.

In many situations CICS is the best choice for implementing client/server application systems. CICS provides an unprecedented spectrum of possibilities:

- Synchronous and asynchronous communication services
- Support for a broad range of protocols
- Distributed data
- Distributed unit of work
- Multiple LUW management options
- Support for external security managers
- Security management in an N-tier environment
- Transaction management and monitoring
- Good performance in a multiuser transaction environment
- Stability of the operating environment
- Homogeneous environment on all CICS platforms.

It is very important in the complicated world of client/server that application systems are implemented in an environment that is as homogeneous as possible. CICS provides almost all of the services that are crucial to a reliable and stable client/server system.

3.1 CICS Configuration

The available CICS configuration options and protocol choices available are reviewed in this section.

3.1.1 Options

VisualAge Generator supports multiple configuration options in a CICS-based client/server communication environment (see Figure 5 on page 26).

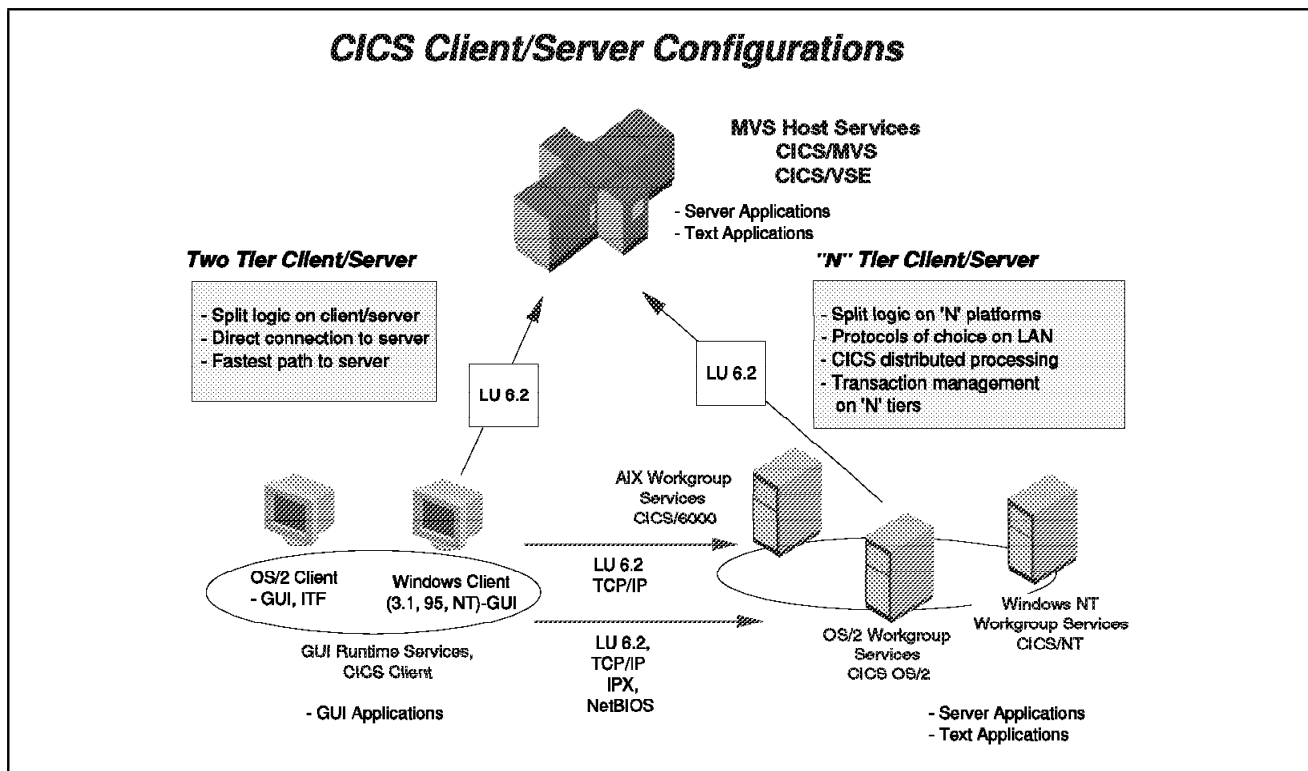


Figure 5. CICS Client/Server Configurations

CICS Clients can provide direct access to CICS servers for two-tier configurations, such as when a GUI client is directly connected to a CICS server. This two-tier approach often provides the best available performance. For a detailed description of CICS Client configurations, see *CICS Clients Unmasked*.

CICS is also ideal for N-tier architectures. This type of configuration allows remote procedure calls (RPCs) to be satisfied at the first CICS platform or to be passed on to another connected CICS platform. Applications can execute on the first server (acting as an application server) or pass through the server (acting as a gateway) to another CICS server by using Distributed Program Link (DPL) support.

Figure 5 shows both two-tier and N-tier configuration options for client calls to server resources.

Notice that ITF is capable of acting as a client in VisualAge Generator client/server communication environments. This allows application calls being tested in ITF to be implemented using the actual executable in the actual target runtime environment. ITF uses the same client/server communication as a GUI client application.

3.1.2 Protocol Choices

Table 2 on page 27 shows the protocol options supported in CICS-based client/server communication configurations.

Table 2. DCE-Based Client/Server Communication Protocol Options						
Platform	MVS CICS Host Services	VSE CICS Host Services	AIX CICS/6000 Workgroup Services	CICS OS/2 Workgroup Services	CICS/NT Workgroup Services	
OS/2 CICS Client	LU 6.2	LU 6.2	LU 6.2 TCP/IP	LU 6.2 TCP/IP IPX NetBIOS	TCP/IP IPX NetBIOS	Two Tier
Windows CICS Client (3.1, 95, NT)	LU 6.2	LU 6.2	LU 6.2 TCP/IP	LU 6.2 TCP/IP IPX NetBIOS	TCP/IP IPX NetBIOS	
AIX CICS/6000 Workgroup Services	LU 6.2	LU 6.2	LU 6.2 TCP/IP	LU 6.2 TCP/IP	LU 6.2 TCP/IP	Three Tier
CICS OS/2 Workgroup Services	LU 6.2	LU 6.2	LU 6.2 TCP/IP	LU 6.2 TCP/IP IPX NetBIOS	LU 6.2 TCP/IP IPX NetBIOS	
CICS/NT Workgroup Services	LU 6.2	LU 6.2	LU 6.2 TCP/IP	LU 6.2 TCP/IP IPX NetBIOS	LU 6.2 TCP/IP IPX NetBIOS	

Protocol options between CICS Client and server platforms provide two-tier support. Connections between CICS server platforms provide N-tier support.

3.1.3 Processing Flow

Figure 6 shows the basic processing flow for a CICS-based client/server communication configuration.

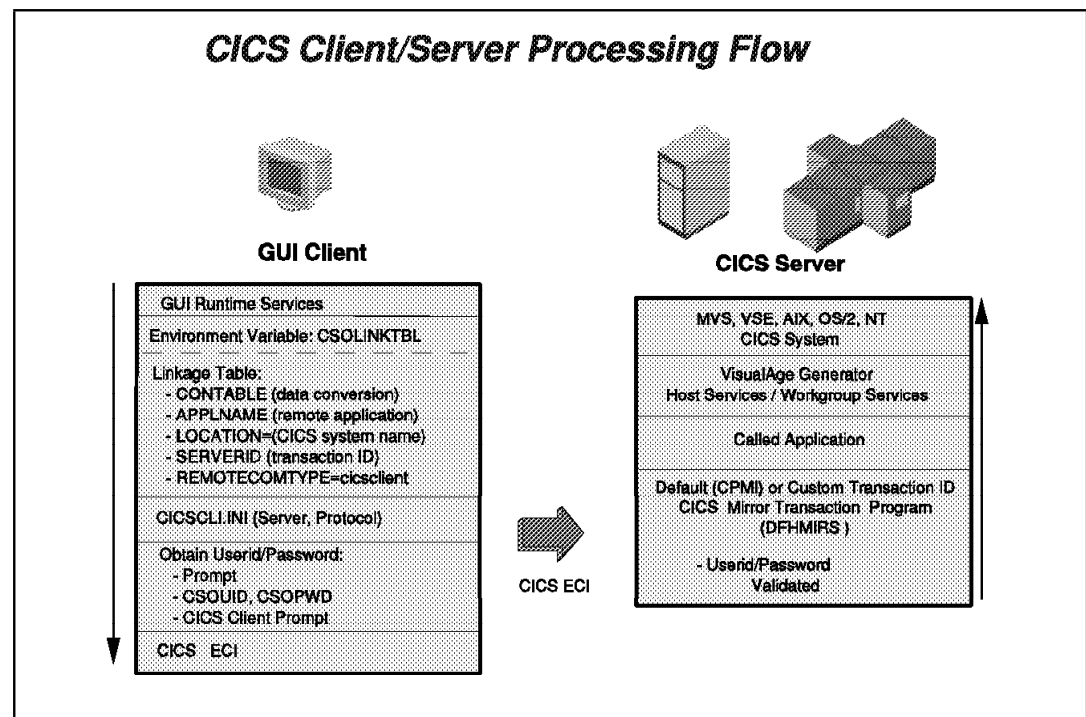


Figure 6. CICS Client/Server Communication Processing Flow

The VisualAge Generator client uses the environment variable CSOLINKTBL to identify the linkage table that will be used to determine how the remote call will be implemented.

The LOCATION parameter is used to identify the CICS server that will receive the application request. (This CICS server platform could pass the request on to another CICS server using DPL support.)

The SERVERID parameter identifies the transaction ID to execute on the CICS server. If none is provided, the default transaction CPMI is used. The choice of a transaction ID can affect security processing as performed by the CICS server. On host environments the transaction ID chosen can affect the database connection and authorization that may be required during the execution of the server application. You may need to identify a transaction ID that is also defined in the CICS resource control table (RCT) so that the appropriate DB2 plan is used for server processing.

User ID and password information is obtained, based on the active VisualAge Generator client/server communication configuration, and is packaged as part of the call to the identified CICS server platform. The method of obtaining the user ID and password information will depend on the setting of the CSOUEXIT environment variable, the version of CICS Client software, and the target CICS server platform.

CICS Client processing uses the CICSCLI.INI file to find information about how the remote procedure call to the requested CICS server will be implemented. Protocol and destination options are identified in the CICSCLI.INI file.

Finally, control is passed from the CICS Client to the CICS server via the ECI interface.

On the server side, CICS authorization is performed as required in the CICS server connection configuration. This can be based on the transaction ID associated with the server request. The transaction-invoked CPMI (or user-defined if a SERVERID value is specified in the linkage table) executes the DFHMIRS CICS-supplied catcher program which, in turn, starts the called application specified in the APPLNAME parameter of the active linkage table entry used for this call.

The LUW can be committed either at the server or by the client, which depends on the application design and the linkage table entry being used.

3.2 CICS Client/Server Scenarios

VisualAge Generator client applications, be they GUI or text clients, can access servers on one or more CICS server platforms. The supported CICS server platforms are these:

- CICS OS/2
- CICS NT
- CICS/6000
- MVS CICS
- VSE CICS

VisualAge Generator client applications can access CICS servers using CICS Client support from these client platforms:

- OS/2
- Windows NT
- Windows 95
- Windows 3.11

In this section, we discuss the configuration of:

- VisualAge Generator Workgroup Services on the three workstation CICS platforms (OS/2, Windows NT, and AIX)
- VisualAge Generator client applications that use CICS Client support for client/server communication.

Figure 7 on page 29 shows this configuration.

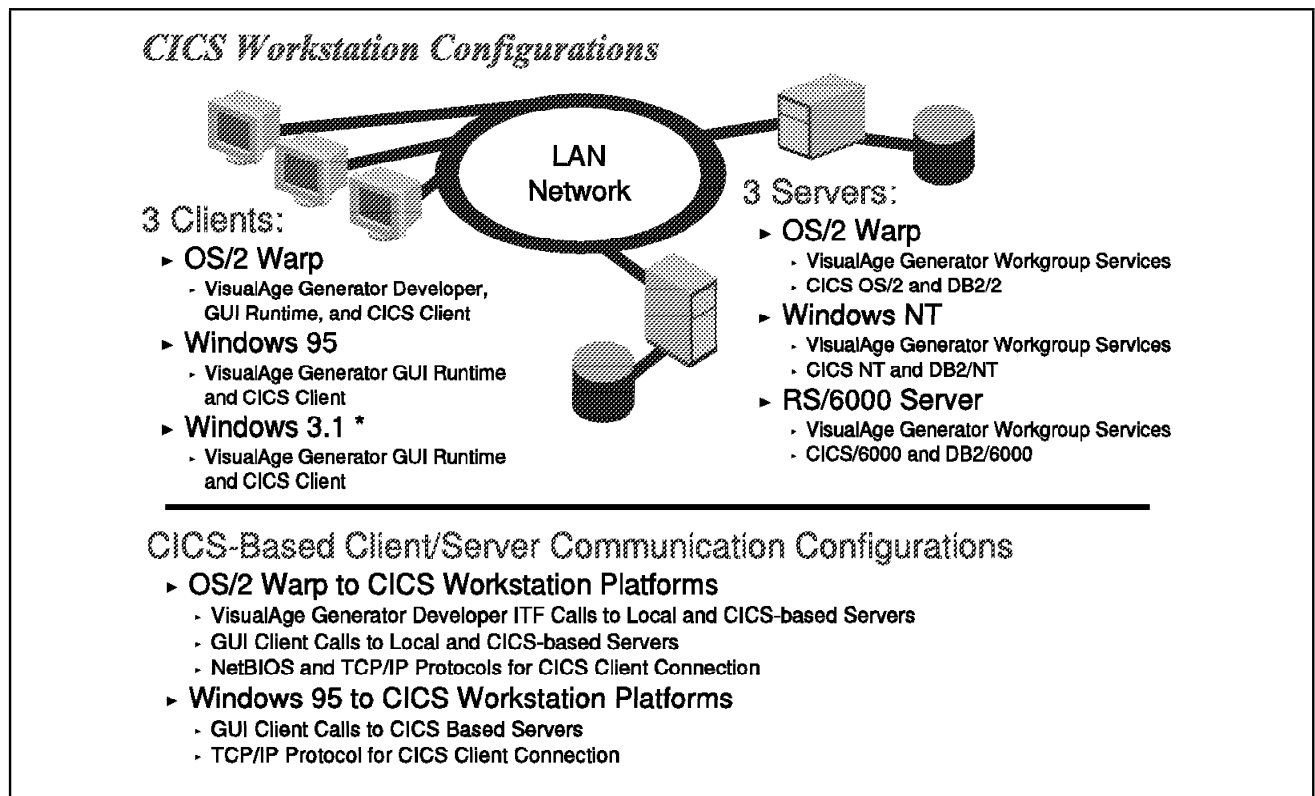


Figure 7. CICS Configuration Implemented during Residency Project

Configuring CICS servers can be a complex task, depending on the target operating system. However, configuring CICS-based client/server communication support for VisualAge Generator clients is very easy using the CICS Client software.

This is one of the reasons we recommend the use of CICS-based client/server communication support (as opposed to VisualAge Generator middleware options for CICS targets).

3.2.1 CICS OS/2 Server

The CICS OS/2 server workstation we set up provided support for:

- CICS OS/2 transaction execution
- Generation of VisualAge Generator GUI client and server applications
- COBOL compilation
- C++ compilation
- Local DB2/2 database access
- CICS Client connections.

We wanted to support CICS Client connections to this server platform using both NetBIOS and TCP/IP options. The output of generation activity was shared with other workstations in our configuration using LAN Server. Generation processing is discussed in Appendix A, "Sample Applications" on page 163.

3.2.1.1 Software

The primary software installed on our CICS OS/2 server platform is listed in Table 3

Table 3. Primary Software Installed on CICS OS/2 Server Workstation	
OS/2 WARP Connect with WIN-OS2 Version 3.00 Component ID 562267200 Current CSD level: IPU8000 Prior CSD level: IPU8000	IBM OS/2 LAN Requester WIN-OS2 Version 4.00 Component ID 562246101 Current CSD level: IP08000 Prior CSD level: IP08000
IBM TCP/IP Version 3.0 for OS/2 Version 3.00 Component ID 562281300 Current CSD level: UN00001 Prior CSD level: UN00001	IBM CICS for OS/2 Version 3.00 Component ID 33H206100 Current CSD level: UN00000 Prior CSD level: UN00000
IBM VisualAge for COBOL for OS/2 Version 1.10.1 Component ID 562279300 Current CSD level: IWZ1101 (1) Prior CSD level: IWZ1100	VisualAge C++ Compiler Version 3.00 Component ID 562201703 Current CSD level: CTC300 Prior CSD level: CTC300
IBM DB2 for OS/2 Single-User Version 2.10 Component ID 562204401 Type 32-bit Current CSD level: WR08000 Prior CSD level: WR08000	IBM OS/2 User Profile Management - Extended Version 4.00 Component ID 562246105 Current CSD level: IP08000 Prior CSD level: IP08000
IBM VisualAge Generator Developer for OS/2 Version 2.02.00 Component ID 562258000 Current CSD level: 0000001 (2) Prior CSD level: 0000001	IBM VisualAge Generator Workgroup Services Version 2.02.00 Component ID 562258500 Current CSD level: 0000001 (2) Prior CSD level: 0000001
Note: 1. We had to apply service, termed CSD1, to IBM VisualAge COBOL before it would work for VisualAge Generator application preparation. 2. We installed the <i>Refresh</i> versions of VisualAge Generator Developer and Workgroup Services. This would be equal to the generally available (GA) version of VisualAge Generator V2.2 with FixPak 1 applied. We highly recommend that you start with at least this level of V2.2.	

The installation of each of the products shown in Table 3 automatically updates the OS/2 CONFIG.SYS as required for basic product operation. Any additional updates to CONFIG.SYS that we made are identified in 3.2.1.2, "Configuration and Customization."

3.2.1.2 Configuration and Customization

To finish setting up our CICS OS/2 server workstation, we completed the following tasks:

- CICS OS/2 customization
- VisualAge Generator Workgroup Services CICS configuration
- Sample application generation and preparation
- CICS OS/2 transaction definition.

These tasks are reviewed in detail below. You should review the appropriate chapters in *Installing VisualAge Generator Workgroup Services* before beginning this activity.

CICS OS/2 Customization: We performed the following tasks to implement a customized CICS OS/2 environment for our sample application servers.

1. Start CICS OS/2 to begin customization.
2. Define a private CICS OS/2 System Initialization Table (SIT).
Changes to CICS OS/2 should be made using a private SIT and not that shipped with CICS OS/2.

Note: We highly recommend that you make all changes to SIT values using your own SIT entry. If you make a mistake and damage the SIT, you can start CICS OS/2 using the CICSGRP environment variable to load only the default SIT provided by CICS OS/2. This will allow you to modify your damaged SIT and recover.

Use the CICS OS/2 CEDA transaction to edit the default SIT. Choose the option to add a new SIT and enter custom values for group name and description. These values are entered on page 1 of the SIT entry, as shown in Figure 8 on page 32. We chose a SIT name of VGWGSSIT. (All three SIT definition pages are shown in Figures 8 through 10.)

3. Define a unique CICS OS/2 system name and application ID.

Each CICS OS/2 region should have a name and application ID other than the default (CICS and CICSOS2) as they exist after the initial installation. We chose the name of CWGS and the application ID of CICSWGS2. This is defined on page two of the SIT, as shown in Figure 9 on page 32.

4. Define protocols supported for CICS Client access.

The CICS OS/2 SIT controls which protocols are supported for CICS Client access and how many connections can be active for each protocol. The number of connections is enforced for NetBIOS, but this number does not seem to be enforced for TCP/IP connections.

NetBIOS connection support seems to allocate the memory required for each possible session during CICS OS/2 startup. (There is a delay in the CICS OS/2 monitor window when the *FAA5570I NetBIOS Listener starting for CICSWGS2 on adapter 0* message is displayed.) protocols supported for CICS Client access.

TCP/IP support seems to dynamically add the memory required for each connection. This suggests that you consider using only TCP/IP connections if you have the option and are concerned about memory use on the CICS OS/2 server platform.

We want to support both NetBIOS and TCP/IP for client connections. We defined a NetBIOS listener adapter of 0, which means the first token ring card in the system, and support for three NetBIOS system connections. For TCP/IP, we defined a host name of *, which means the active TCP/IP host name, a port designation of 1435, and support for three TCP/IP system connections. Our CICS OS/2 server host name is ITSCSRV1. These definitions are also on page 2 of the SIT as shown in Figure 9 on page 32.

5. Enable UPM-based security.

We chose to use UPM as the security manager for our CICS OS/2 server. Three security manager options are available with CICS OS/2:

UPM	OS/2 User Profile Manager. UPM delivered with CM/2, DB2/2, and upgraded if LAN Requestor or LAN Server is installed.
SNT	Sign-on Table. CICS security scheme.
NONE	Private security. <i>Do not choose this option unless you have a working security exit configured for CICS OS/2.</i> The choice of NONE does not mean no security, it means that CICS OS/2 will ask your configured CICS security exit (FAAEXP07) to make access decisions. If you do not have a security exit configured after you have selected the NONE option and shut down CICS OS/2, you cannot start CICS OS/2 again until: • A working security exit exists. • An alternative SIT is used to start CICS OS/2.

The SIT entry selecting UPM as the security manager is shown in Figure 10 on page 33.

6. Shut down CICS OS/2 with the CQIT transaction or by closing the CICS OS/2 monitor window.

FAASIT2		System Initialization Table-1		More : +	
Group Name.	VGWSSIT				
Description	CICS OS/2 SIT FOR VG/WGS2				
System Sizes					
CWA size.	0				
Maximum TWA size.	1024				
Trace table size.	500				
Task Control					
Maximum number of tasks	6	(1-99)			
Minimum free tasks.	2	(0-99)			
Task Classes.	1 2 3 4 5 6 7 8 9 10				
Maximum tasks in Class.	1 1 1 1 1 1 1 1 1 1	(0-99)			
Default Process Priority.	86	(0-255)			
CICS System Priority.	0	(0-255)			

Figure 8. CICS OS/2 SIT—Page 1. The values for group name and description were defined as part of CICS OS/2 customization. The maximum transaction work area (TWA) size was altered as part of the tasks defined for VisualAge Generator Workgroup Services CICS configuration.

FAASIT3		System Initialization Table-2		More : - +	
Group Name.	VGWSSIT				
System Communications					
Local System ID	CWGS				
Local System Appl ID.	CICSWG2				
Default Remote System ID.					
NetBIOS Support					
NetBIOS Listener Adapter.	0	(0, 1 or B)			
Maximum NetBIOS Systems	3	(0-254)			
TCP/IP Support					
TCP/IP Local Host Name.	*				
TCP/IP Local Host Port.	1435	(* or 1-65535)			
Maximum TCP/IP Systems.	10	(0-999)			
PNA Support					
Load PNA Support.	N	(Y or N)			
PNA Model Terminal.					

Figure 9. CICS OS/2 SIT—Page 2

FAASIT4		System Initialization Table-3		More : -	
Group Name. : VGWGSIT					
Security					
Security Manager.		UPM (SNT, UPM or NONE)			
Miscellaneous					
Initial Transaction ID.		CLOG			
Dump on Abend		N		(Y or N)	
Date Format		MMDDYY		(ddmmyy,mmddyy,yyymmdd)	
External File Manager Name.					
User Conversion Table					
EBCDIC code page.		00037		(0-65535)	

Figure 10. CICS OS/2 SIT—Page 3

VisualAge Generator Workgroup Services CICS Configuration: This activity integrates VisualAge Generator Workgroup Services with CICS and sets up the control files that support preparation processing and starting CICS OS/2 with Workgroup Services support. Refer to the appropriate topics in *Installing VisualAge Generator Workgroup Services* before beginning this activity.

1. Customize Workgroup Services ELAENV.CMD (environment setup command file)

The ELAENV.CMD is called as part of the ELARUNC.CMD that starts CICS OS/2 with Workgroup Services support. ELAENV.CMD is also called as part of VisualAge Generator preparation processing for applications that will run in a CICS OS/2 environment.

Note: If you review the internals of the ELAENV.CMD you will see that it has logic which prevents it from running twice (see Figure 11).

```

/*-----*/
/* Determine if this program has already been run in this environment. */
/*-----*/
if GetValue(execname||'_RUN') = '' then
do

:
'&WGS ELAENV Logic'

:
end /* end-if this program has not been run */

exit

```

Figure 11. Control Logic in ELAENV.CMD to Limit to One Pass Execution

If you are not aware of the bypass logic in the ELAENV.CMD and change the command settings to correct problems discovered during preparation, you may not realize why your changes do not seem to take effect. You need to open a new OS/2 window or reset the ELAENV_RUN environment variable to nulls so that the logic of ELAENV.CMD does not immediately force an exit.

We made the following changes to the settings in the ELAENV.CMD that control where products are installed and which COBOL product (IBM or Micro Focus) we are using in our environment (see Figure 12 on page 34).

```

os2_install_drive   = 'C:'
ela_install_drive   = 'D:'
cics_install_drive  = 'D:'
cobol_install_drive = 'D:'

ela_install_dir     = '\VGWGS2'
cics_install_dir    = '\CICS300'
cobol_install_dir   = '\IBMCOBOL'

default_ezernls     = 'ENU'
default_database    = 'SAMPLE'

ela_bit_mode        = '32'
ela_cics_group      = 'VGWGS'

cics_version        = '3.0'

call_cicsenv        = 1

user_work           = 'E:\VGCS\GENOUT\OS2CICS'

cobol_type          = 'IBM'

```

Figure 12. Control Settings in ELAENV.CMD

2. Change the LIBPATH settings in CONFIG.SYS for IBM COBOL.

Because we have chosen to use IBM COBOL, we will be running generated VisualAge Generator applications in 32-bit mode in CICS OS/2. During the installation of VisualAge Generator Workgroup Services, there is no way to determine which COBOL runtime environment, Micro Focus COBOL (16-bit) or IBM COBOL (32-bit) that you will use at runtime.

You need to ensure that D:\VGWGS2\LIB\32 and not D:\VGWGS2\lib\16. is in the settings for LIBPATH.

If you do not correct the CONFIG.SYS LIBPATH setting, you will get the following CICS error when you use VisualAge Generator programs:

```
FAA5513E Process 'id' terminated abnormally, RC = 'rc'
```

3. Customize Workgroup Services ELARUNC.CMD.

You can customize settings in the ELARUNC.CMD file to have a VisualAge Generator communications gateway (used by VisualAge Generator middleware support) started as part of the starting CICS OS/2. You can also have the Micro Focus CICS OS/2 option called instead of calling CICS OS/2 directly.

We did not make either of these optional choices, so these control settings remained unchanged.

We did ask that CICS OS/2 be started with a cold start to ensure that any previous abends were cleaned up completely (we were testing quite a bit of code). This was done by modifying the parameters passed to the ELARUNC.CMD file at invocation. We added /C as a parameter in the *Start CICS OS/2 with IBM VisualAge Generator Support* program icon.

4. Start CICS OS/2 using the *Start CICS OS/2 with IBM VisualAge Generator Support* icon in the Workgroup Services folder.

The steps that remain are done inside CICS OS/2.

5. Define Workgroup Services to CICS OS/2 by importing VisualAge Generator supplied definitions using the CAIM transaction.

The following steps are required:

- a. Back up the file FAAAEFIE.BTR in the \CICS300\RUNTIME\DATA directory.
- b. Copy the file ...\\VGWGS2\ELAEX300.BTR to the \CICS300\RUNTIME\DATA directory as file name FAAAEIE.BTR.
- c. Run the CAIM transaction to import the contents of the FAAAEIE.BTR file. This file contains the VisualAge Generator Workgroup Services definitions used in CICS OS/2.

The CAIM transaction menu options suggested in the section "Importing the VisualAge Generator Workgroup Services CICS Resources" of the *Installing VisualAge Generator Workgroup Services* did not work for us when using CICS OS/2 V3.

The following discussion, extracted from an online discussion about CICS OS/2, tells us why and what to do instead.

Subject: Problems with CAIM import in CICS OS/2 3.0

After successfully installing CICS OS/2 3.0 we tried to import a group definition using the CAIM transaction. We have placed a FAAAEFIE.BTR file in the C:\CICS300\RUNTIME\DATA directory. During import we keep getting error FAA1315. This tells us that the FAAEFIE file is unavailable.

What do we do now?

Subject: Problems with CAIM import in CICS 3.0

I read a previous append regarding the FAA1315W error code when issuing CAIM transaction. As suggested I tried to enable the FAAAEFIE.BTR file using CEMT.

The file became enabled but when I try to execute the CAIM transaction I get the same error. When I checked again with with "CEMT I FILE" transaction I see that the FAAAEFIE.BTR file was closed and disabled again.

I also tried an approach where I used the CEDA transaction to modify the file definition for FAAAEFIE.BTR and set the open option to yes.

When I restarted CICS the file was open and enabled but when I issued the CAIM transaction I got the same error.

Do I miss something?

Subject: Problems with CAIM import in CICS 3.0

If my memory serves me correctly, this "feature" was added just before CICS OS/2 V3.0 was shipped.

I believe you can still import and export groups resetting the file open and enabled each time.

We expect to fix this "feature" in CSD1.

Apologies for the inconvenience.

Subject: Problems with CAIM import in CICS 3.0

Until you get the fix in CSD1 one way to work with the file closing problem is not to use the 'Backup' option in CAIM. If you use CEMT to enable the file, then run CAIM, the import will work as long as the backup option is not used.

After the import has run it will be necessary to use CEMT to enable the file.

So we used the CEMT transaction to ensure that the FAAAEIE.BTR file was enabled and then used the CAIM transaction to import the VGWGS group. This file contains the VisualAge Generator Workgroup Services definitions.

We did not import the EZERSMP group. There is an empty/missing file referenced in the EZERSMP group that causes messages to be displayed when starting CICS OS/2. We did not want to use the VisualAge Generator sample applications, so by not importing the EZERSMP group we avoid triggering these warning messages.

6. Update customized SIT to set the Transaction Work Area (TWA) to 1024
VisualAge Generator transactions in a CICS environment use TWA resources. CICS must be told to reserve the number of bytes used by VisualAge Generator CICS applications. The SIT entry for TWA is visible in Figure 8 on page 32.
7. Shut down CICS OS/2 with the CQIT transaction or by closing the CICS OS/2 monitor window.
8. Start CICS OS/2 using the *Start CICS OS/2 with IBM VisualAge Generator Support* icon in the Workgroup Services folder.
9. Verify that VisualAge Generator Workgroup Services is installed and functioning using the ELAM transaction.

The programs behind the ELAM transaction were written in VisualAge Generator. Running the ELAM transaction exercises Workgroup Services functions and proves that VisualAge Generator Workgroup Services was installed and customized properly.

We go one step further by entering an invalid option on the selection menu shown by the ELAM transaction. This forces Workgroup Services to look up an error message in the error message table associated with the application. When you see the message *ELA00304A Type a valid selection number, then Press Enter.* you have forced message-table to be accessed.

Note: During our testing of incomplete servers, we triggered hard abends in the CICS OS/2 environment. Occasionally, strange behavior would occur after such an abend. If we had *broken* VisualAge Generator Workgroup Services support in CICS OS/2 none of our applications would function. The ELAM transaction would continue to show a menu. But, if we entered an invalid option on the ELAM menu, an unexpected and unformatted error messages would show up on the terminal screen and the CICS OS/2 monitor window. This told us that Workgroup Services was broken. To fix it, we had to cold start CICS by adding /C as an option to CICS invocation.

10. Set up database support for VisualAge Generator applications.

The default DB2/2 database that will be accessed by a VisualAge Generator application running in CICS OS/2 (or on OS/2) is controlled by VisualAge Generator using these environment variables:

EZERSQLDB Defines the default database.
ELARTRDB_tttt Defines the database to be used for a specific CICS transaction where tttt is the transaction ID.

We could either set the value for EZERSQLDB in CONFIG.SYS or use the ELAENV.CMD file option for setting the default database name for VisualAge Generator Workgroup Services applications. We defined a default database name of SAMPLE in the ELAENV.CMD file (see Figure 12 on page 34).

11. Enable database access for VisualAge Generator applications.

CICS OS/2 does not use the user ID and password information obtained from the ECI call to support DB2/2 database access. This means an active local UPM logon (as opposed to a LAN Requestor logon) must exist to support DB2/2 database access for all programs running in CICS OS/2.

If no valid local logon (with the appropriate DB2/2 database access) is active when a VisualAge Generator application that contains SQL access is started, a Local Logon dialog is shown on the CICS OS/2 server workstation. This logon request is triggered by the **SQL CONNECT TO database** logic contained in all VisualAge Generator applications that contain SQL statements. The connect is issued before an SQL statement is issued—it will be issued even if the logic path does not use any of the SQL statements contained in the application.

If nobody is at the server to enter a valid user ID and password when the local logon dialog is shown, the server will wait and eventually time out. We recommend that you add a local logon for DB2/2 access to the STARTUP.CMD for the workstation or the ELARUNC.CMD used to start CICS OS/2.

Sample application generation and preparation: Two sample application servers were generated for the CICS OS/2 target runtime platform. They support remote calls from VisualAge Generator clients using either CICS-based or VisualAge Generator Middleware-based client/server communication support.

We have two servers that are part of the sample application that must be generated for CICS OS/2: VGC2OS2 and VGC3OS2. Server VGC2OS2 will be called by the sample application GUI client. Server VGC3OS2 will be called by a sample application server when a three-tier server call to CICS OS/2 is requested. The generation process is as follows:

1. Define linkage table for CICS OS/2 sample application servers.

We recommend that the remote linkage convention (linktype=remote) be used for any CICS-based server. This allows the server to be called from either a local or remote application. The calling application could be a client (text or GUI) or just another server application running either in CICS or on some other runtime platform.

Figure 13 shows the linkage table entries we used to generate the VGC2OS2 and VGC3OS2 applications.

```
:CALLLINK APPLNAME=VGC2OS2 LIBRARY=VGC2OS2 REMOTECOMTYPE=cicsclient
  PARMFORM=commdata LINKTYPE=remote LUWCONTROL=server SERVERID=VGCS.

:CALLLINK APPLNAME=VGC3OS2 LIBRARY=VGC3OS2 REMOTECOMTYPE=cics
  PARMFORM=commdata LINKTYPE=remote LUWCONTROL=server SERVERID=VGCS.
```

Figure 13. Linkage Table for CICS OS/2 Sample Application Servers

During generation, the critical options are PARMFORM and LINKTYPE. We can change the REMOTECOMTYPE value used during the actual call from a client or non-CICS server and still reach these CICS server applications.

2. Define generation options for CICS OS/2 and IBM COBOL.

We used the default options file provided by VisualAge Generator (EFKOPDFT.OPT) as a starting point but we did add one option:

```
/COBOL=IBM
```

We need this option since we are using IBM COBOL. The default, for compatibility with previous releases of VisualAge Generator, is to generate Micro Focus COBOL if no COBOL option is specified.

If you do not request IBM COBOL and you do not have Micro Focus COBOL installed, you get a message telling you that COBCLI.LBR cannot be found during application preparation (compile process).

3. Generate CICS OS/2 sample application servers (VGC2OS2 and VGC3OS2)

Figure 14 contains the generation commands we used to generate the sample application servers used in CICS OS/2.

```
EZE2GEN GENERATE VGC2OS2 /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%%EZEENV%  
/linkage=h:\vgcs\applst.lkg /system=OS2CICS  
/trace=stmt,sqlio >e:\vgcs\genout\vgc2os2.log  
  
EZE2GEN GENERATE VGC3OS2 /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%%EZEENV%  
/linkage=h:\vgcs\applst.lkg /system=OS2CICS  
/trace=stmt,sqlio >e:\vgcs\genout\vgc3os2.log
```

Figure 14. Generate Statements for CICS OS/2 Sample Application Servers

We included the /trace generation option so that we could trace the execution of the server applications in CICS OS/2. This option is not recommended for production servers, because of the heavy overhead associated with an application that includes trace support. You must turn trace support on before the applications are traced in a CICS environment. The VisualAge Generator Workgroup Services ELAZ transaction is used to start trace support for VisualAge Generator applications. See *Installing VisualAge Generator Workgroup Services* for guidance on using the ELAZ transaction.

Our generation output directory is the same as the user_work directory identified in the ELAENV.CMD file (see Figure 12 on page 34).

Because of this, we must lower CICS OS/2 before preparation. If we do not stop CICS OS/2 and we have called previous versions of the servers, preparation will fail because the dynamic link library (DLL) has been loaded and is locked by the CICS OS/2 process. The following message results:

```
EZE4238I The COBOL compile is starting for member VGC2OS2  
PP 5622-793 IBM VisualAge for COBOL for OS/2 1.1 in progress ...  
End of compilation 1, program VGC2OS2, no statements flagged.  
ILink : fatal error LNK1083: cannot open run file - The file is open.  
EZE4240E A COBOL compiling error occurred for member VGC2OS2
```

If we had used a different generation-output directory, we would not have problems with locking. However, we would have to copy the DLLs to a directory referenced by the user_work setting in the ELAENV.CMD file or by the CICSWRK environment variable. The user_work directory value is added to the CICSWRK environment variable by the ELAENV.CMD logic.

CICS OS/2 Transaction Definition: To directly run a program in CICS, you must have a transaction definition. The transaction definition identifies the program that should be started when the transaction ID is entered.

When a program is run in CICS OS/2 using the External Callable Interface (ECI) programming interface you can identify a transaction ID to be used to support the requested program. VisualAge Generator supports the definition of a transaction value in the ECI call by using the SERVERID linkage table option.

By default, CICS OS/2 uses the transaction CPMT when no transaction ID is identified as part of the ECI call. The CPMT transaction will start a program named DFHMIR. This mirror program issues a CICS LINK to the program requested using the ECI interface. The process is as follows:

- Define a private transaction for sample application servers called using the CICS OS/2 ECI.

The CPMT transaction is part of CICS. We typically do not modify CICS-provided definitions but instead use our own.

Use the CEDA transaction to add a new Program Control Table (PCT) entry named VGCS. The VGCS transaction must still start the program DFHMIR. This is because we specify the VGCS transaction as part of the ECI call created by VisualAge Generator when a client requests a server using the REMOTECOMTYPE=cicsclient SERVERID=VGCS linkage table entry option (see Figure 13 on page 37).

3.2.2 CICS/6000 Server

The CICS/6000 server workstation we set up provided support for

- CICS/6000 transaction execution
- Preparation of VisualAge Generator server applications
- Local DB2/6000 database access
- CICS Client connections.

We wanted to support CICS Client connections to this server platform using the TCP/IP protocol option. The generation activity performed on the CICS OS/2 server, which also supports VisualAge Generator application generation, directs files to this AIX workstation for preparation processing.

3.2.2.1 Software

The primary software installed on our CICS/6000 server platform is listed in Figure 15.

AIX	V4.1.3
CICS for AIX	V2.1.0
ENCINA server	V2.1.0
Workgroup Services for AIX	V2.2.0
C Set++ for AIX	V3.1.3
DB2/6000	V2.1.0
DB2/6000 options installed:	
- DDCS	
- DB2 Server, Single User and Client Enabler	
- Communications support:	
DRDA AS, IPX, TCP/IP, SNA	

Figure 15. Software Installed to Support CICS/6000 Server

3.2.2.2 Install Process for VisualAge Generator Workgroup Services for AIX

If an earlier version of VisualAge Generator Workgroup Services is already installed, it must be removed before installing a new version.

For example, if VisualAge Generator Workgroup Services Version 2.0 is installed, it can be removed using the following command if you are logged in as user root:

```
installp -u vgws20.obj
```

To install the new VisualAge Generator Workgroup Services image, use the following command:

```
installp -ad vgws22v.img all
```

The VisualAge Generator Workgroup Services image file `vgwgs22v.img` is specific to the version of AIX, CICS/6000, and DB2/6000 installed on our RS/6000 workstation. If you have different base software, you need to use a different installation image for VisualAge Generator Workgroup Services. For details, refer to Chapter 10, "Installing VisualAge Generator Workgroup Services for AIX" in the *Installing VisualAge Generator Workgroup Services*.

3.2.2.3 Customization for Running Applications on AIX/CICS Server

After the installation of the required software, you must configure CICS/6000 and VisualAge Generator Workgroup Services.

Note: You have to be logged in as root before performing the actions described in this section.

Configuring CICS/6000 Support for CICS Client TCP/IP Connection: TCP/IP is a popular and flexible protocol that can be used to support CICS Client connections to a CICS/6000 region.

To enable TCP/IP connection support you must define a listener for your CICS/6000 region and add a new service name to the TCP/IP configuration. The configuration that we describe is based on our CICS/6000 region named *vgsmc*. The process is as follows:

1. Create a listener definition (LD).

The listener detects requests to your CICS region and is associated with a TCP/IP port. Use SMIT to create a listener:

```
smit
- Applications
  - Customer Information Control System (CICS) Version 2
    - Manage CICS Regions
      - Define Resources for a CICS Region
        - Manage Resources
          - Listeners
            - Add Listener
```

Fill in the following fields:

```
Listener Identifier      = CICSLD
Region name             = vgsmc
Update, Install or Both = Update
Group to which resource belongs = cics
Activate resource at cold start = yes
Protocol type           = TCP
TCP Adapter address     = ""
TCP Service name        = ""
```

The TCP Adapter Address and the Service name are left blank. This means we are using the defaults; CICS can use any of the TCP/IP adapters on the machine and uses the 1435 service port. You can also specify a service:

```
TCP Service name      = cicstcp1
```

This name has to be added to the `/etc/services` file.

2. Add a service name to the `/etc/services` file (if required)

When you have specified a service name in the LD, you must add this name with a unique port number to the file. For example,

```
cicstcp1    1435/tcp    # TCP listener for region vgsmc
```

You can find more information about CICS/6000 TCP/IP connections in Chapter 3, “Network Configuration for CICS on Open Systems” in the *CICS on Open Systems Intercommunication Guide*.

Customize CICS/6000 to Support VisualAge Generator Workgroup Services: This activity teaches CICS about VisualAge Generator Workgroup Services, including control files that support preparation processing and starting CICS OS/2 with Workgroup Services support. You should refer to the appropriate topics in *Installing VisualAge Generator Workgroup Services* before beginning this activity.

Customization has two steps:

1. Set the environment variables in the environment file for the CICS/6000 region.

The environment file can be found in the directory `/var/cics_regions/regionName`. For the SAMPLE database on AIX, we use the following settings:

```
DB2INSTANCE=db2
EZERNLS=ENU
FCWDB2DIR=/usr/lpp/db2_02_01
FCWDBUSER=db2
FCWDBPASSWORD=DB2
EZERSQLDB=SAMPLE
FCWLIBPATH=/u/vgcsres/genout
FCWDPATH=/u/vgcsres/genout
LIBPATH=/usr/lpp/vgwgs22/lib
```

The VisualAge Generator sample application servers used in CICS/6000 are physically located in the home directory `genout`. User ‘db2’ is the owner of the DB2 instance named *db2*. When you want to switch on the trace option, you have to set these variables also:

```
FCWTROPT=31
FCWTROUT=/var/cics_regions/vgsmc/data/FCWTRACE.OUT
```

For additional configuration guidance please review *Installing VisualAge Generator Workgroup Services*.

2. Define Workgroup Services to CICS/6000 using the definitions supplied by the VisualAge Generator.

Add Workgroup Services programs, transactions and other definitions to the CICS for AIX permanent and runtime databases, by running the following commands for region ‘vgsmc’:

```
export CICSREGION=vgsmc
fcwcicsinstall
cicsinstall -r vgsmc -g VGWGS
```

CICS/6000 and VisualAge Generator Workgroup Services have to reference the same DB2 object library *db2.o*. See “Create a DB2 Shared Object” on page 42 for a description of how to accomplish this task.

3.2.2.4 Application Generation and Preparation for CICS/6000 Server

Before you start generating, you must

- Create a DB2 shared object.
- Create an AIX user and customize the profile.
- Check REXEC authority.

To generate the application, you must

- Define the appropriate options to support preparation on AIX.
- Create a linkage table.

After generation, make the applications active by defining them to CICS.

Create a DB2 Shared Object: CICS transactions and Workgroup Services both refer to the DB2 shared object at run time. This DB2 object library is needed in the preparation of the VisualAge Generator applications.

- To create the DB2 shared object, issue the following commands:

```
cd /user/lpp/db2_02_01/lib
ar -vx libdb2.a
mv shr.o db2.o
```

- To create the symbolic links,
 1. Run the *db2ln* script which can be found in /usr/lpp/db2_02_01/cfg.
 2. Then issue this command:

```
ln -s /usr/lpp/db2_02_01/lib/db2.o /usr/lib/db2.o
```

Create an AIX User and Customize the User Profile: To support VisualAge Generator application preparation for the CICS/6000 target runtime environment you need to be able to transfer the generated C++ source and the preparation command file to the required RS/6000 workstation. To do this, create a user that has all the required environment variables defined in the user profile:

1. Use SMIT to create an AIX user named *vgcsres*.
2. Add the following lines to the *vgcsres* user profile:

```
# Set the DB2 variables for VG preparation
export DB2INSTANCE=db2
export FCWDB2DIR=/usr/lpp/db2_02_01
export PATH=/u/db2/sql1lib/bin:$PATH
```

As an alternative, you can also execute the profile of the DB2/6000 instance that you want to use. Add the following line to the user's profile:

```
. /home/db2/sql1lib/db2profile
```

User 'db2' is the owner of the DB2/6000 instance.

Setup REXEC Authority to Support Preparation Processing: The generated C++ source code has to be transferred to an AIX directory and then the preparation script will be executed. To perform this action, the OS/2 generation machine must have remote execution (REXEC) authority. Execute the following command on an OS/2 prompt of the generator machine to see whether or not it has REXEC authority:

```
rexec vgrisc -l vgcsres -p secret ls
```

You need to specify a hostname, a user ID and password. In our example, these are 'vgrisc,' 'vgcsres,' and 'secret,' respectively. If you have the authority, the *ls* (the AIX *dir* command) will be executed.

The hostname, user ID, and password are used later as values for these generation options: /DESTHOST, /DESTUID, and /DETPASSWORD.

Generation Options for AIX CICS: The following preferences in the options file are used for the generation of our sample application:


```

/SYSTEM=AIXCICS
/CICSENTRIES=RDO
/COMMENTLEVEL=1
/GENHELPMAPS
/GENMAPS
/GENTABLES
/CHECKTYPE=ALL
/LOCVALID
/NOSQLVALID
/SQLDB=SAMPLE
/DBUSER=db2
/DBPASSWORD=db2
/DESTHOST=vgrisc.raleigh.ibm.com
/DESTUID=vgcsres
/DETPASSWORD=vgrisc
/DESTDIR='/home/vgcsres/genout'
/LISTINGONERROR
/GENOUT=e:\vgcsres\GENOUT\%EZEPIID%\%EZEENV%

```

Linkage Table for AIX CICS: The linkage table for generating remote applications for CICS/6000 used in the sample application is shown in Figure 16.

```

:CALLLINK APPLNAME=VGC2AIX LIBRARY=VGC2AIX REMOTECOMTYPE=cicsclient
  PARMFORM=commdata LINKTYPE=remote LUWCONTROL=server SERVERID=VGCS.

:CALLLINK APPLNAME=VGC3AIX LIBRARY=VGC3AIX REMOTECOMTYPE=cics
  PARMFORM=commdata LINKTYPE=remote LUWCONTROL=server SERVERID=VGCS.

```

Figure 16. Linkage Table for CICS/6000 Sample Application Servers

Figure 17 contains the generation commands we used to generate the sample application servers used for CICS/6000.

```

EZE2GEN GENERATE VGC2AIX /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%EZEENV%
  /linkage=h:\vgcs\applst.lkg /system=AIXCICS
  /OPTIONS=h:\vgcs\vgcaix.opt
  /trace=stmt,sqlio >e:\vgcs\genout\VGC2AIX.log

EZE2GEN GENERATE VGC3AIX /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%EZEENV%
  /linkage=h:\vgcs\applst.lkg /system=AIXCICS
  /OPTIONS=h:\vgcs\vgcaix.opt
  /trace=stmt,sqlio >e:\vgcs\genout\VGC3AIX.log

```

Figure 17. Generate Statements for CICS OS/2 Sample Application Servers

Preparation starts automatically unless you specify the /NOPREP option.

Defining Programs to CICS/6000: The generation process also creates a script that defines the generated application in CICS. It adds a program definition and a transaction definition to the CICS tables. When you do not want to define transactions for your server programs, all programs can run under the CPMI mirror transaction. However, for main applications (text clients) that run in CICS/6000, you are required to specify a transaction. Otherwise you have no way to start the application!

The CICS script for an application is named '`<applicationName>c.scr`'. A script for adding application VGC2AIX to CICS is shown in Figure 18 on page 44.

```

#!/bin/ksh
eval "cicsadd -r $CICSREGION -c pd -P VGC2AIX RSLKey=public PathName="/home/vgcsres/genout/VGC2AIX.d11" "
eval "cicsadd -r $CICSREGION -c td -P VGCS RSLKey=public ProgName=VGC2AIX "

```

Figure 18. VisualAge Generator Script to Add Application Definitions to CICS/6000

The generated script does not contain the flag 'RSLKey=public'. We added this flag to set the authority to call the program and transaction remotely. We also changed the transaction ID to VGCS.

To run the script, you must first change its attributes:

```
chmod +x vgc2aixc.scr
```

Then you can execute the script by typing

```
vgc2aixc.scr
```

Now the applications are defined in CICS. To make them active, shut down and restart CICS.

3.2.2.5 Operations in CICS/6000

Formal CICS/6000 education or experience is required to administer and operate a production CICS/6000 environment. Two common operations are discussed here.

Shutting Down and Restarting CICS/6000: There are special considerations when you use a DCE, Encina SFS, and CICS/6000 configuration. Shutting down and restarting CICS is clearly described in the CICS/6000 publications.

Problems when Starting or Restarting CICS/6000: Sometimes a CICS/6000 region will not restart because it is locked. This is typically due to an incorrect shutdown. To unlock CICS/6000, issue the following command on an AIX prompt (as user root):
 cicsrlck -r <regionName> For example, type cicsrlck -r vgsmc

3.2.3 CICS Client

A VisualAge Generator GUI client application can use CICS-based client/server communication support when CICS Client software is available on the client workstation. To have a GUI client application call a server located on one of the supported CICS server platforms, the following is required:

- A linkage table with the REMOTECOMTYPE=cicsclient option.

The client/server communication option used by a GUI application to call a remote server is determined dynamically. In linkage table syntax, this is termed REMOTEBIND. GUI applications and the VisualAge Generator ITF always perform runtime binding. That is, GUI applications and the ITF decide how the call to a server is to be made (when the call is about to be made), based on the :CALLLINK linkage table entry for the remote server application.

For a GUI application, the active linkage table is defined by the setting of the CSOLINKTBL environment variable. The contents of the linkage table are read into memory when the VisualAge Generator GUI runtime support code (EZE2RUN or EZERUN) is started.

Linkage table processing when using the ITF to access generated servers is discussed in 2.1, "Calling Generated Applications" on page 19.

- A valid working CICS Client configuration (defined in the CICSCLI.INI file) that points to the required CICS server.

CICS Client definitions are made very simply. A CICSCLI.INI configuration file can include multiple server definitions. CICS Client software supports ECI calls and remote CICS terminals. The terminal support is very useful for testing the function of a server definition. When the terminal can connect to the server, you are ready to test the actual ECI call.

- Control over which CICS server is to be called.

The CICSCLI.INI file can contain multiple active CICS server definitions. When you start a CICS terminal, you are asked to select one of the available server definitions. By default, the first definition is used for an ECI call.

You can use the LOCATION=systemname linkage table option to name the target CICS server. The systemname value corresponds to the Server = xxxxxxx value in the CICSCLI.INI configuration file. If you think multiple CICS servers might be accessed using CICS Client software, include the LOCATION option in your linkage table definition.

- A valid working server in the target CICS system.

Setup of the workstation CICS server system setup is discussed in 3.2.1, “CICS OS/2 Server” on page 29. The target server must be generated with a linkage table with the LINKTYPE=remote and the PARMFOM=commdata options.

3.2.3.1 Option File and Linkage Table for GUI Application Generation

We used the default option file (EFKOPDFT.OPT) for GUI application generation. The GUIs in our sample application were generated for both the OS/2 and Windows client platforms (see Figure 19).

```
EZE2GEN GENERATE VGPING /MSL=VGPINGZ
      /GENOUT=e:\VGCS\GENOUT\OS2      /linkage=h:\vgcs\applst.1kg
      /system=OS2gui >e:\vgcs\genout\VGPING.log

EZE2GEN GENERATE VGPINGSD /MSL=VGPINGZ
      /GENOUT=e:\VGCS\GENOUT\OS2      /linkage=h:\vgcs\applst.1kg
      /system=OS2gui >e:\vgcs\genout\VGPINGsd.log

EZE2GEN GENERATE VGHELP /MSL=VGPINGZ
      /GENOUT=e:\VGCS\GENOUT\OS2      /linkage=h:\vgcs\applst.1kg
      /system=OS2gui >e:\vgcs\genout\vghelp.log

EZE2GEN GENERATE VGPING /MSL=VGPINGZ
      /GENOUT=e:\VGCS\GENOUT\wingui   /linkage=h:\vgcs\applst.1kg
      /system=wingui >e:\vgcs\genout\VGPING.logNT

EZE2GEN GENERATE VGPINGSD /MSL=VGPINGZ
      /GENOUT=e:\VGCS\GENOUT\wingui   /linkage=h:\vgcs\applst.1kg
      /system=wingui >e:\vgcs\genout\VGPINGsd.logNT

EZE2GEN GENERATE VGHELP /MSL=VGPINGZ
      /GENOUT=e:\VGCS\GENOUT\wingui   /linkage=h:\vgcs\applst.1kg
      /system=wingui >e:\vgcs\genout\VGHELP.logNT
```

Figure 19. Sample Application Generation Commands for GUI Client. Embedded GUI applications are generated as a result of the /GENEMBEDDEDGUIS default generation option.

Sample application servers are generated with support for remote calls. Generation of the servers is discussed in the platform-specific CICS server topics:

- 3.2.1, “CICS OS/2 Server” on page 29
- 3.2.2, “CICS/6000 Server” on page 39.

GUI client applications must also be generated with remote call support for servers that will be called remotely. If the LINKTYPE=remote linkage table option is not used for a called application when the GUI application is generated, the call is implemented as a local call. Local calls from a GUI application are not affected by a runtime linkage table definition.

The linkage table option of REMOTEIND=runtime is always used by a GUI application, but only for calls generated as remote. Figure 20 shows the linkage table entries used for generation for the remote CICS workstation-based servers that are called by the GUI client in our sample application.

```
:CALLLINK APPLNAME=VGC2OS2  LIBRARY=VGC2OS2  PARMFORM=commdata
      LUWCONTROL=server LINKTYPE=remote.

:CALLLINK APPLNAME=VGC2AIX  LIBRARY=VGC2AIX  PARMFORM=commdata
      LUWCONTROL=server LINKTYPE=remote.
```

Figure 20. Linkage Table used to Generate Sample Application GUI Client for CICS Servers

The LOCATION and SERVERID :CALLLINK options are not required at generation time.

3.2.3.2 CICS Client Configuration

We configured CICS Client access for CICS server systems on three workstations. The definitions provided terminal and ECI support. Once working, VisualAge Generator GUI applications and VisualAge Generator ITF can use these connections with the appropriate :CALLLINK linkage table entries.

CICS Client configurations, as defined in the CICSCLI.INI file, are very similar for OS/2, Windows 95, and Windows NT. Figure 21 on page 47 contains the server definition section of a CICSCLI.INI file.

```

;*****
;* IBM CICS Client - Initialization File
;*****
;
;
;
;-----
; Server section - This section defines a server to which the client may
;                   connect. There may be several Server sections.
;
;
Server = CICSNETB           ; Arbitrary name for the server
Description = NetBIOS Server ; Arbitrary description for the server
Protocol = NETBIOS         ; Matches with a Driver section below
NetName = CICSWGS2         ; The server's NetBIOS name
Adapter = 0                ; Use NetBIOS on LAN adapter 0
UpperCaseSecurity = Y      ; Fold Userid and Password to uppercase
InitialTransid = CLOG      ; Initial terminal transaction name

Server = CICSWGS2           ; Arbitrary name for the server
Description = TCP/IP Server ; Arbitrary description for the server
Protocol = TCPIP           ; Matches with a Driver section below
NetName = itscsrv1.almaden.ibm.com ; The server's TCP/IP address
Port = 1435                ; Use the default TCP/IP CICS port

Server = CICSAXR           ; Arbitrary name for the server
Description = CICS/6000 RTP ; Arbitrary description for the server
Protocol = TCPIP           ; Matches with a Driver section below
NetName = vgrisc.raleigh.ibm.com ; The server's TCP/IP address
Port = 1435                ; Use the default TCP/IP CICS port
;

```

Figure 21. CICSCLI.INI Definitions for Workstation CICS Servers. Only a portion of the full CICSCLI.INI file is shown. The definitions for the communication drivers used for each supported protocol are given at the bottom of the full CICSCLI.INI file. See the CICSCLI.INI file included as part of a CICS Client installation on your selected target operating system for additional descriptive information and CICS Client communication driver configuration.

Multiple protocol options are supported when connecting CICS Client with CICS server systems. The protocol you select may be based on existing company standards, local preferences, or compatibility with other products used on the end-user's client workstation.

CICS Client protocol support is reviewed in 3.1.2, "Protocol Choices" on page 26. For additional guidance on CICS Client configuration options, please consult CICS Client documentation and *CICS Clients Unmasked*, SG24-2534-01.

3.2.3.3 GUI Client Configuration and Execution

Control of server location, client/server communication configuration, and methods of obtaining user ID/password data used as part of the call in some client/server communication options are determined by the runtime settings for GUI application execution.

The following settings and commands configure VisualAge Generator GUI runtime support options and start the GUI client application:

SET CSOLINKTBL=d:\filename

Name of linkage table to be read to support runtime binding of remote calls

SET CSOUEXIT=userauth

Name of user authentication routine that provides user ID and password values used in client/server communication

SET CSOUID=uuuuu and SET CSOPWD=ppppp

Environment variables used as source of user ID and password values if the default (CSOUIDPW) user authentication exit is used.

SET CSOTROPT=n

Sets client/server communication trace options. A value of two traces all client/server calls.

SET CSOTROUT=d:\filename

Defines location of client/server communication trace output file.

EZE2RUN OPEN guiappl

Starts GUI application.

Figure 22 shows the linkage table entries used at run time (referenced by CSOLINKTBL) for the remote CICS workstation-based servers that are called by the GUI clients in our sample application.

```
:CALLLINK APPLNAME=VGC2OS2 LIBRARY=VGC2OS2 PARMFORM=commdata
LUWCONTROL=server LINKTYPE=remote REMOTECOMTYPE=cicsclient
SERVERID=VGCS LOCATION=CICSWG2.

:CALLLINK APPLNAME=VGC2AIX LIBRARY=VGC2AIX PARMFORM=commdata
LUWCONTROL=server LINKTYPE=remote REMOTECOMTYPE=cicsclient
SERVERID=VGCS LOCATION=CICSAIXR.
```

Figure 22. Linkage Table used for Runtime GUI Client Calls to CICS Servers. The REMOTECOMTYPE=cicsclient option identifies the use of CICS-based client/server communication support. The LOCATION option is used to identify the target CICS system name as defined in the CICS Client configuration file. The SERVERID option identifies the CICS transaction ID that should be used as part of the ECI call. If SERVERID is not used, the default transaction ID of CPMI is used to support the request.

When we run the sample application and call a server using CICS-based client/server communication support, if we use the CSOTROPT=2 environment variable setting, we then can see in the CSOTRACE.OUT file how the processing is implemented (see Figure 23).

```
<Jan 15 09:43:53>->CMINIT
<Jan 15 09:43:53><-CMINIT - 0.029291 s
<Jan 15 09:43:53>->CMCALL
<Jan 15 09:43:53> Calling application VGC2OS2
<Jan 15 09:43:53> ->readFromLinkTbl
<Jan 15 09:43:53> <-readFromLinkTbl - 0.207975 s
<Jan 15 09:43:53> ->loadAndInitDriver
<Jan 15 09:43:53> <-loadAndInitDriver - 0.323233 s
<Jan 15 09:43:53> ->CMDV_INIT
<Jan 15 09:43:53> ->CICSCLIENT:CMDV_INIT
<Jan 15 09:43:53> <-CICSCLIENT:CMDV_INIT - 0.028159 s
<Jan 15 09:43:53> <-CMDV_INIT - 0.084962 s
<Jan 15 09:43:53> ->CICSCLIENT:CMDV_CALL
<Jan 15 09:43:59> CMDV_CALL: LuwType=SERVER
<Jan 15 09:43:59> ->CICSCLIENT ECI_CALL
<Jan 15 09:44:06> <-CICSCLIENT ECI_CALL - 7.567954 s
<Jan 15 09:44:06> <-CICSCLIENT:CMDV_CALL - 14.583481 s
<Jan 15 09:44:06><-CMCALL - 15.405744 s
```

Figure 23. CSOTRACE.OUT Entries for GUI Client Call to CICS OS/2 Server

See Appendix A, “Sample Applications” on page 163 for information on using the generated GUI client application.

3.2.3.4 VisualAge Generator Developer ITF as a Client

In ITF, a called batch application (server) can be interpreted or called as a generated application when a linkage table is identified in the ITF general preferences profile.

When the ITF calls a generated application, the generated application acts very much like a GUI client application. A valid client/server communication configuration is required.

Calling generated applications from ITF can impact database identification and authorization processing (see Chapter 2, “Testing Applications with VisualAge Generator Developer ITF” on page 19).

3.2.4 CICS-to-CICS Connections

Any CICS server platform can connect to any other CICS server platform. Connections between CICS OS/2 and MVS CICS are reviewed in Chapter 6, “DBCS-Enabled Client/Server Configurations” on page 139. In this section we review the CICS definitions required to support remote program calls from CICS/6000 to CICS OS/2 and from CICS OS/2 to CICS/6000 using TCP/IP-based connections.

3.2.4.1 Implement Support for TCP/IP in CICS OS/2

To implement a CICS/6000 to CICS OS/2 connection that uses TCP/IP, ensure that CICS OS/2 is configured to support TCP/IP-based connections. The CICS OS/2 SIT entries required to support TCP/IP are discussed as part the CICS OS/2 configuration task Define on page 31 and shown in Figure 9 on page 32.

3.2.4.2 Implement a CICS OS/2 Connection Definition in CICS/6000

To support the call from a program in CICS/6000 to a program in CICS OS/2, you have to connect the two systems by enabling the selected communication protocol (TCP/IP) and identifying the target system. To establish TCP/IP communications support for CICS/6000 you create a listener definition. We used the same listener definition we used to support a CICS Client TCP/IP connection to CICS/6000 (see “Configuring CICS/6000 Support for CICS Client TCP/IP Connection” on page 40). A CICS/6000 communications definition is required to point to the target CICS OS/2 system (see Figure 24 on page 50).

New Communication Identifier	[WGS2]
Communication Identifier	WGS2
Region name	vgsmc2
Update Permanent Database OR Install OR Both	Update
Group to which resource belongs	[]
Activate the resource at cold start?	yes
Resource description	[Communications Definition]
Number of updates	9
Protect resource from modification?	no
Connection type	cics_tcp
Name of remote system	[CICSWGS2]
SNA network name for the remote system	[]
SNA profile describing the remote system	[]
Default modename for a SNA connection	[]
Gateway Definition (GD) entry name	[]
Listener Definition (LD) entry name	[CICSLD]
TCP address for the remote system	[itscsrv1.almaden.ibm.com]
TCP port number for the remote system	[1435]
DCE cell name of remote system	[././]
Timeout on allocate (in seconds)	[0]
Code page for transaction routing	[IBM-037]
Set connection in service?	yes
Send userids on outbound requests?	sent
Security level for inbound requests	verify
UserId for inbound requests	[VGCS]
Transaction Security Level (TSL) Key Mask	[none]
Resource Security Level (RSL) Key Mask	[none]
Transmission encryption level	none

Figure 24. CICS/6000 Communications Identifier for CICS OS/2 System

The program on the remote CICS OS/2 system must also be defined to CICS/6000 as a remote program (see Figure 25).

New Program Identifier	[vgc3os2]
Program Identifier	vgc3os2
Region name	vgsmc2
Update Permanent Database OR Install OR Both	Update
Group to which resource belongs	[itso]
Activate resource at cold start?	yes
Resource description	[Program Definition]
Number of updates	1
Protect resource from modifications?	no
Program enable status	enabled
Remote system on which to run program	[WGS2]
Name to use for program on remote system	[VGC3OS2]
Transaction name on remote system for program	[VGCS]
Resource Level Security Key	[public]
Program path name	[]
Program type	program
User Exit number	[0]
Is a user conversion template defined?	no
Is this a program that should be cached?	no

Figure 25. CICS/6000 Program Definition for Remote CICS OS/2 Program

With the connection implemented and the remote program defined, you can use the VGPING sample application to implement a call from the GUI client to the tier-2 server VGC2AIX and then to the tier-3 server VGC3OS2. Figure 26 on page 51 shows the VGPING sample application GUI with the results after such a call.

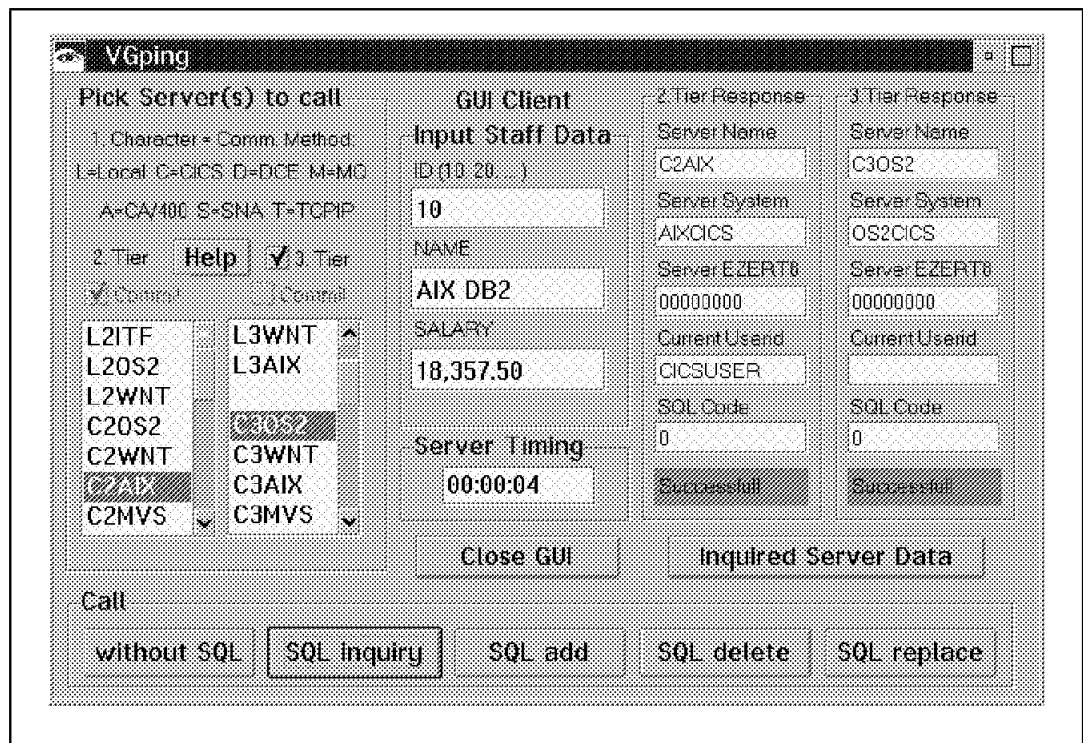


Figure 26. VGPING Sample Application GUI after Three-Tier Call: GUI to CICS/6000 to CICS OS/2

3.2.4.3 Implement a CICS/6000 Connection Definition in CICS OS/2

To support the call from a program in CICS OS/2 to a program in CICS/6000, you have to connect the two systems by enabling the selected communication protocol (TCP/IP) and identifying the target system. TCP/IP communications support for CICS OS/2 is defined in the SIT. The TCP/IP definitions in our CICS OS/2 SIT can be seen in Figure 9 on page 32. A CICS OS/2 connection and session table (TCS) entry is required to point to the target CICS/6000 system (see Figure 27).

FAATCS5		Connection and Session Table	
Connection Name.	CAIX		
Group Name	VGCS		
Connection Type.	TCP	(APPC, NETB or TCP)	
Connection Priority.	086	(0-255)	
Description.	DPL TO VGRISC AT RTP		
Session Details			
Session Count.	10	(1-99)	
Session Buffer Size.	40000	(512-40000)	
Attach Security.	V	(L=Local, V=Verify)	
Partner Code Page.	00037		
TCP/IP Details			
Local host name.	*		
Remote host name	VGRISC.RALEIGH.IBM.COM		
Remote host port	*	(1-65535, or *)	

Figure 27. CICS OS/2 Communications Identifier for CICS/6000 System

The program on the remote CICS/6000 system must also be defined to CICS OS/2 as a remote program (see Figure 28 on page 52).

FAAPPT2	Processing Program Table-1
Program Name	: VGC3AIX
Group Name	: VGCS
Program Type(P,M,D).	: P (P, M or D)
Resident(P,T,N).	: N (P, T or N)
System ID.	: CAIX
Remote Program Name.	: VGC3AIX
Remote Transaction ID.	:
Description.	: DPL TO RTP VGRISC FOR 3-TIER

Figure 28. CICS OS/2 Program Definition for Remote CICS/6000 Program

With the connection implemented you can use the VGPING sample application to implement a call from the GUI client to the tier-2 server VGC2OS2 and then to the tier-3 server VGC3AIX. Figure 29 shows the VGPING sample application GUI with the results after such a call.

The screenshot shows the VGPING application window with the following sections:

- Pick Server(s) to call:**
 - 1. Character = Conn. Method: L=Local C=CICS D=DCE M=MQ A=CAV400 S=SNA T=TCPIP
 - 2. Tier: ☐ 2 Tier ☒ 3 Tier
 - 3. Control: ☒ Control ☐ Control
 - Server lists: L2ITF, L2OS2, L2WNT, C2WNT, C2AIX, C2MVS, L3WNT, L3AIX, C3OS2, C3WNT, C3MVS
- GUI Client Input Staff Data:**
 - ID (10, 20, ...): 10
 - NAME: DB22 SJC
 - SALARY: 18,357.50
 - Server Timing: 00:00:03
- 2 Tier Response:**
 - Server Name: C2OS2
 - Server System: OS2CICS
 - Server EZERT8: 00000000
 - Current Usend: USERID
 - SQL Code: 0
 - Status: Successful
- 3 Tier Response:**
 - Server Name: C3AIX
 - Server System: AIXCICS
 - Server EZERT8: 00000000
 - Current Usend: CICSUSER
 - SQL Code: 0
 - Status: Successful
- Call:**
 - without SQL
 - SQL inquiry**
 - SQL add
 - SQL delete
 - SQL replace

Figure 29. VGPING Sample Application GUI after Three-Tier Call: GUI to CICS OS/2 to CICS/6000

Chapter 4. DCE-Based Client/Server

In some situations, DCE is the best choice for implementing client/server application systems. DCE provides support for:

- A broad range of protocols
- An open security manager
- Distributed data
- Security management in an N-tier environment
- Single logon.

DCE may be viewed by your organization as a strategic choice for implementing client/server communication support. VisualAge Generator, by providing support for DCE-based client/server communication, allows you to combine the ease of use benefits of VisualAge Generator's support for client/server development and implementation with an open and strategic choice for client/server communication.

4.1 DCE Configuration

Support for DCE-based client/server communication is new with VisualAge Generator Version 2.2. Two- and three-tier client/server configurations are supported by VisualAge Generator using DCE client/server communication support.

4.1.1 Options

VisualAge Generator supports multiple configuration options in a DCE-based client/server communication environment. DCE-based client/server communication supports the use of VisualAge Generator clients with servers generated for native execution on the AIX, OS/2, and Windows NT workstation operating systems. DCE also provides support for calls to servers running on the MVS CICS and IMS/VS host transaction platforms and the VM/ESA host environment (see Figure 30 on page 54).

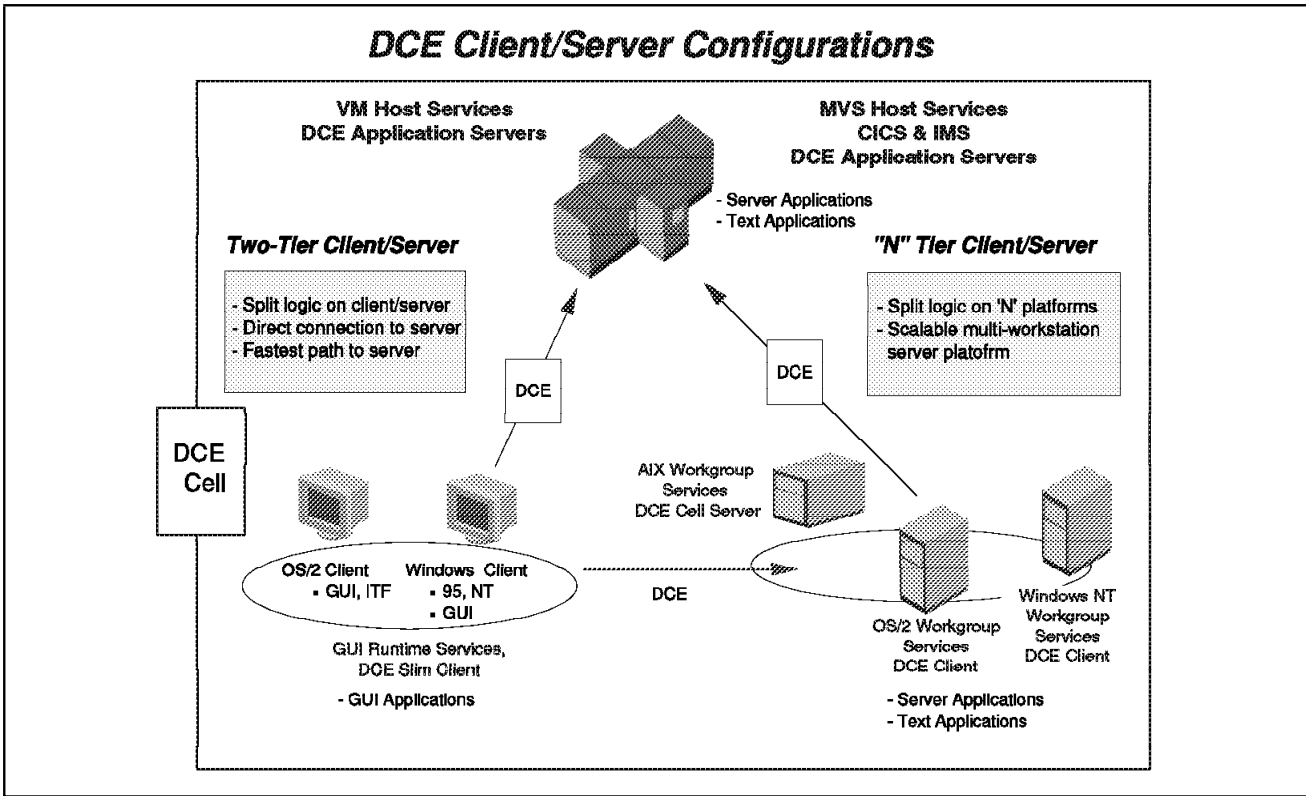


Figure 30. DCE-Based Client/Server Configurations

4.1.2 Protocol Choices

Table 4 shows the protocol options supported in DCE-based client/server communication configurations.

Table 4. DCE-Based Client/Server Communication Protocol Options					
Platform	MVS Host Services (CICS and IMS)	AIX Workgroup Services (native or CICS)	OS/2 Workgroup Services	Windows NT Workgroup Services	
OS/2 Client	TCP/IP	TCP/IP	TCP/IP or NetBIOS (1)	TCP/IP	Two Tier
Windows Client (3.1, 95, NT)	TCP/IP	TCP/IP	TCP/IP	TCP/IP or IPX (2)	
AIX Workgroup Services (native or CICS)	TCP/IP	TCP/IP	TCP/IP	TCP/IP	Three Tier
OS/2 Workgroup Services	TCP/IP	TCP/IP	TCP/IP or NetBIOS (1)	TCP/IP	
Windows NT Workgroup Services	TCP/IP	TCP/IP	TCP/IP	TCP/IP or IPX (2)	
Note:					
(1)	NetBIOS is supported by IBM Directory and Security Server for OS/2 Warp Version 4.0				
(2)	IPX is supported by Gradient PC-DCE/32 for Windows 95, NT				

4.1.3 DCE-Based Client/Server Communication Processing

Figure 31 shows how client and server platforms interact with the DCE Cell server.

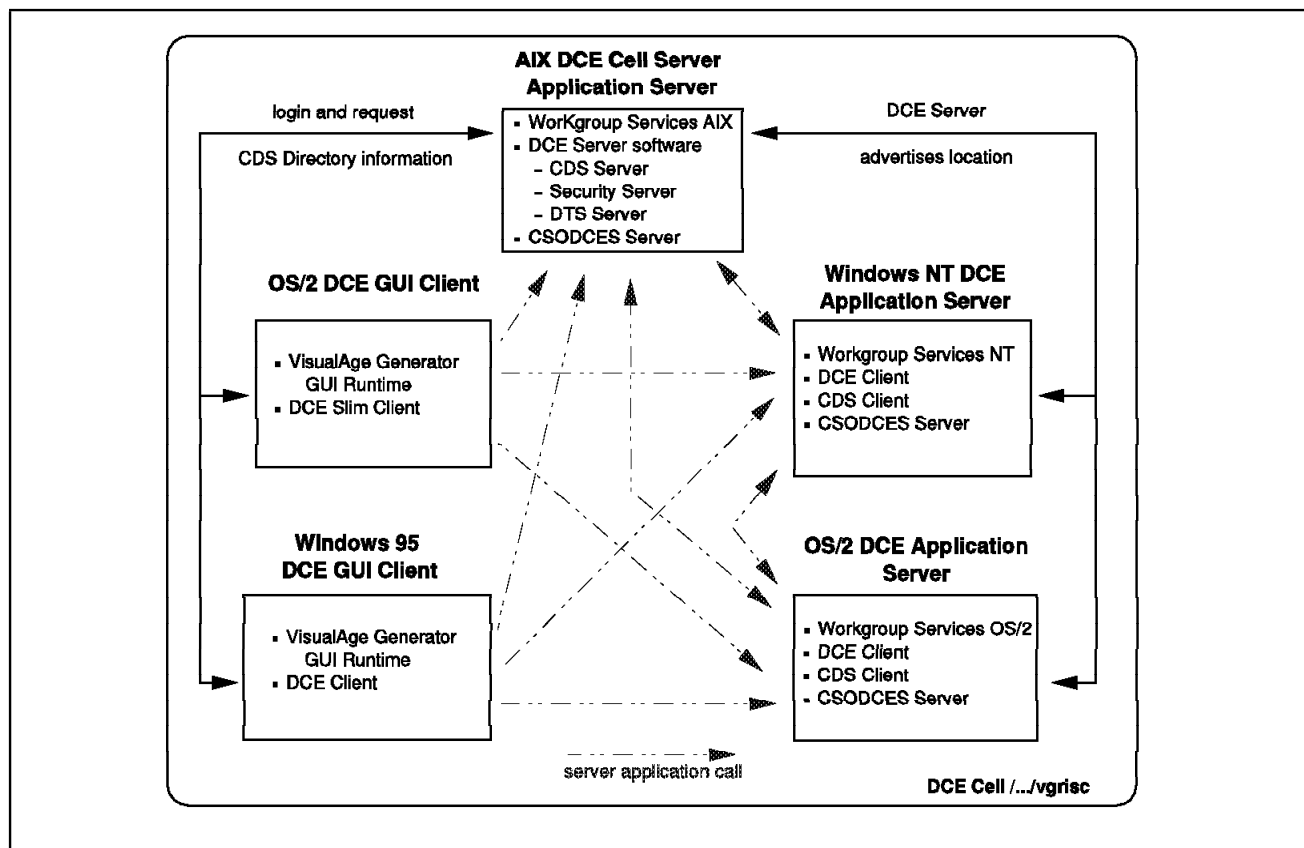


Figure 31. Distributed Operation in a DCE Cell

The processing steps performed to support DCE-based client/server communication, as shown in Figure 31, are described below:

1. The DCE cell server is started on the AIX workstation.
2. The DCE Client is started on each application server workstation. This establishes an interface to the active DCE cell.
3. The VisualAge Generator DCE server program, CSODCES, is started on each application server workstation. The parameters for CSODCES are:

-c or -d This is the optional cleanup parameter. The -c value is the documented default.¹³

configuration-file name

The configuration file specifies which VisualAge Generator server applications are available (can be called) on the application server.

¹³ There is a problem with VisualAge Generator V2.2 at the Fixpak 3 level. The documented default parameter (-c) must be used to trigger the related processing. If -c is not provided, the CSODCES program runs as if the -d parameter was provided, which makes the -d parameter the effective default. The -d parameter, when used, causes an error message (Could not open data file, -d).

The DCE server program CSODCES requires a DCE logon with the authority required to write entries in the DCE CDS. The DCE logon principal (user) and password are provided in the key table (keytab) on the server workstation.

4. A list of the VisualAge Generator applications available on the application server is advertised in the DCE cell by the CSODCES program. This list is stored using the CDS function of the DCE cell. This information (a list of available VisualAge Generator server applications by DCE server workstation name) is now available for reference by all (authorized) DCE clients.
5. The DCE Client is started on the client workstation. Information about servers defined in the DCE cell is shared with the client (CDS data).

If secure client/server communications are configured (REMOTECOMTYPE=dcsecure) and the appropriate DCE authorizations are defined, the end user must perform a DCE logon.

Once DCE Client software has been started and a DCE logon established, VisualAge Generator client applications can call VisualAge Generator servers applications using DCE RPC support.
6. The client application issues calls to server applications using DCE-based client/server communication support. The request is passed to the DCE server program CSODCES. If the REMOTECOMTYPE=dcsecure option was selected, the DCE server program validates the client request with the security server function of the DCE cell.

4.2 DCE Client/Server Scenarios

VisualAge Generator support for DCE-based client/server communication allows a call to a server application from a client to be implemented using DCE-enabled RPC support. Location control and any required data type conversions are controlled through the :CALLLINK definitions in the active linkage table.

We implemented DCE-based client/server communication scenarios that supported GUI client applications on OS/2 and Windows 95, calling server applications on:

- OS/2
- Windows NT
- AIX 4.1.3

The use of DCE-based client/server communication requires that a DCE cell server exist. We used the AIX 4.1.3 workstation as our cell server.

OS/2 or Windows NT could also have been configured as a DCE cell server with the appropriate software installed (see your DCE product documentation). For additional information about implementing a DCE cell review, see

Understanding OSF DCE 1.1 for AIX and OS/2, SG24-4616
DCE Cell Design Considerations, SG24-4746

The setup of these workstation platforms with DCE-based client/server communication support is reviewed in this section.

4.2.1 Configuration Basics

Before we discuss the actual implementation tasks, we review terms, the configuration used in our scenario, and the DCE user definitions used in our environment.

4.2.1.1 DCE-Based Client/Server Communication Terminology

To understand DCE-based client/server communication support, it may be helpful if these terms are defined:

Application Server Workstation

The physical workstation where VisualAge Generator server applications are installed and run when called by clients. Both the DCE server program and the called VisualAge Generator server applications run on this workstation.

DCE Server Program

A program named CSODCES is provided as part of VisualAge Generator Workgroup Services. The program reads identified DCE configuration file to identify the VisualAge Generator server applications that can be called through this instance of the DCE server program.

VisualAge Generator Server Application

Generates the batch application called by VisualAge Generator. Is called on behalf of requesting client applications by the DCE Server program CSODCES.

GUI Client Workstation

The physical workstation where VisualAge Generator GUI client applications are installed and run. Calls to VisualAge Generator server applications are processed using DCE client support.

DCE Client

The software required to support DCE-based client/server communication. DCE Client is required on both the GUI client and application server workstations.

CDS Client

The software required on the application server workstation to support the DCE-based client/server communication.

DCE Cell

The named environment or domain that supports communication between member workstations. VisualAge Generator uses the RPC subset of a DCE environment to implement client/server communication support.

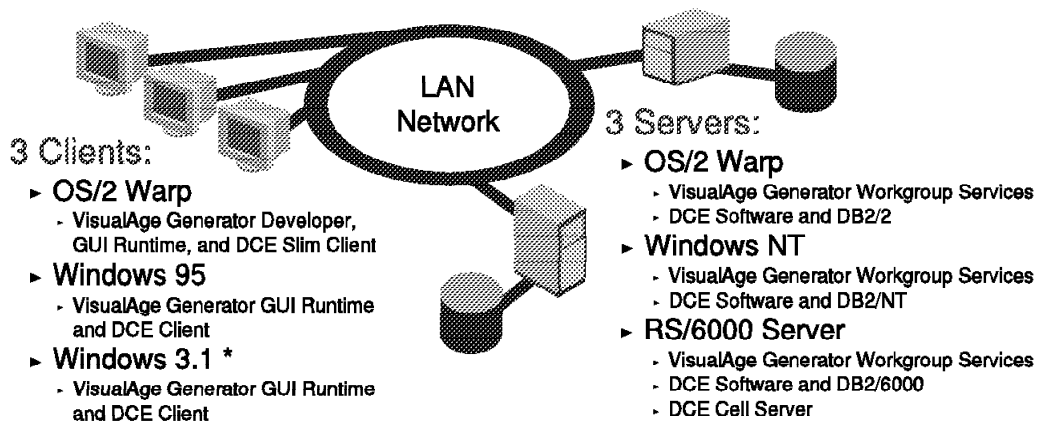
4.2.1.2 Configuration Scenario

Our goal was to implement a DCE-based client/server communication configuration that provided support for these client and server platforms:

- AIX DCE cell server (with application server support)
- OS/2 DCE Application Server Workstation
- OS/2 DCE Client Workstation
- Windows NT DCE Application Server Workstation
- Windows 95 DCE Client Workstation.

Figure 32 on page 58 provides an overview of our DCE configuration.

DCE PS/2 Workstation Configurations



DCE-Based Client/Server Communication Configurations

- ▶ **OS/2 Warp to DCE Enabled Workstation Platforms**
 - VisualAge Generator Developer ITF Calls to Local and Remote DCE-based Servers
 - GUI Client Calls to Local and Remote DCE-based Servers
 - TCP/IP Protocol for DCE Connection
- ▶ **Windows 95 to CICS Workstation Platforms**
 - GUI Client Calls to Remote DCE-based Server
 - TCP/IP Protocol for DCE Connection

Figure 32. VisualAge Generator Client/Server Communication DCE Configuration Overview

All the workstations shown in Figure 32 must have TCP/IP installed and configured to support DCE client/server communication. To test whether TCP/IP is installed and configured correctly, PING the host name of any of the other workstations from any workstation. Each workstation should be able to PING all other workstations that will be used in the DCE configuration.

Figure 33 shows the PING command used to test TCP/IP communication to the AIX DCE cell server.

```
[H:\vgcs]ping vgrisc.raleigh.ibm.com
PING vgrisc.raleigh.ibm.com: 56 data bytes
64 bytes from 9.67.172.228: icmp_seq=0. time=187. ms
64 bytes from 9.67.172.228: icmp_seq=1. time=125. ms
64 bytes from 9.67.172.228: icmp_seq=2. time=125. ms

----vgrisc.raleigh.ibm.com PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 125/145/187
```

Figure 33. PING Command Used to Test TCP/IP Communication between Workstations

Review your TCP/IP documentation for TCP/IP communication configuration guidance.

4.2.1.3 Client and Server User Definitions

Before we implement a DCE-based client/server communication environment we need to determine how DCE user definitions (principals or accounts) will be used to control the environment. We know that:

- A client (end user) DCE logon is not required unless a secure DCE-based call (REMOTECOMTYPE=dcesecure) is required.
- The VisualAge Generator DCE server program (CSODCES) requires a DCE principal with appropriate CDS authority to advertise the available VisualAge Generator server applications.

DCE supports the management of sets of user principals with the organization and group constructs.

We define a DCE principal for each end-user and each application-server workstation. DCE organization and group capabilities are used to define roles and authorities for each set of (client and server) principals.

The following groups and principals are defined as part of an organization named *visgen*:

Group Name	Principals/Role
vgusers	vguser1 vguser2 vguser3 DCE principal IDs for end-users. Used for dce_login on client workstation.
vgusradm	vgadm1 vgadm2 DCE principal IDs for administrative end-users. Used for dce_login on client workstation. Secure DCE-based client/server communication is implemented to support this group of users.
vgservers	vgos2sj1 vgos2sj2 vgaixsj1 vgaixsj2 DCE principal IDs for application server workstations. Used for implicit dce_login, using the principal entry in the keytab on the server workstation, by the DCE server program CSODCES.

4.2.2 AIX DCE Cell Server Workstation

Using AIX for our DCE cell server was an easy choice. DCE cell server support was already installed on our target AIX workstation.

In several of our scenarios, we do not call VisualAge Generator servers that run on AIX. We use the AIX workstation only as the DCE cell server that enables our choice of DCE-based client/server communication for our VisualAge Generator sample application system. We could have implemented DCE cell server support on the OS/2 or Windows NT workstation platforms, or even on a host system if we did not want to use the AIX workstation as the cell server.

We had the following software on our AIX DCE cell server workstation:

- AIX Version 4.1.3
- AIX DCE Version 2.1 (including CDS, Security, and DTS server support)
- DB2 Version 2.1
- VisualAge Generator Workgroup Services for AIX Version 2.2 (Fixpak 3).

We have ignored the other software installed on the AIX workstation as they were not directly involved in this DCE-based client/server communication configuration.

Note: VisualAge Generator Workgroup Services for AIX Version 2.2 was required only when we configured the AIX workstation as a DCE application server workstation. If VisualAge Generator server applications are not called on the AIX workstation, VisualAge Generator Workgroup Services for AIX Version 2.2 is not required to support DCE-based client/server communication.

4.2.2.1 Managing DCE on AIX

This publication does not provide a detailed guide to DCE management and administration. You need DCE product-specific documentation and possibly training to perform this task. We did learn the following:

- Starting DCE on AIX
DCE can be started with the 'rc.dce' command. DCE can also be started by using 'smitty'. Enter

```
smitty dce
```

and then choose the **Restart DCE/DFS Daemons** menu option.

- Checking whether or not DCE is running on AIX
Issue this command on AIX to see if DCE processes are running:

```
ps -ef | grep dce
```

- Stopping DCE on AIX
DCE can be stopped with the dce.clean command or you can use SMIT (or smitty).

4.2.2.2 DCE Environment Configuration

All DCE components of our DCE-based client/server communication configuration will run within a single DCE cell. We used a DCE cell that was already defined on the AIX workstation.

When no DCE cell is defined or you do not want to use an existing DCE cell, you can create a new one. Refer to the appropriate DCE manual for your environment for information about creating a DCE cell.

In the following configuration steps, we assume that you already have defined a DCE cell, a clearinghouse, and a cell administrator *cell_admin*.

To configure DCE-based client/server communication support, perform the following steps:

1. Create DCE organizations, groups, and users.

For these steps you have to logon as *cell_admin*. On AIX, our DCE cell server workstation, you can use the dcecp tool or SMIT to support the configuration tasks.

Note: SMIT (or 'smitty') provides a full-screen, panel-driven interface, which users new to AIX and/or DCE will probably find much easier to use. The line commands shown below are the same ones that smitty ends up issuing. They are displayed for you to see.

Figure 34 on page 61 shows the dcecp commands issued (after entering dcecp on the AIX command line) to create the DCE environment required for our configuration (see 4.2.1.3, "Client and Server User Definitions" on page 59).

```

dcecp> organization create visgen
dcecp> group create vgservers -inprojlist yes
dcecp> group create vgusers -inprojlist yes
dcecp> group create vgusradm -inprojlist yes
dcecp> user create vuser1 -password vuser1 -mypwd xxx -group vgusers -o visgen
dcecp> user create vuser2 -password vuser2 -mypwd xxx -group vgusers -o visgen
dcecp> user create vuser3 -password vuser3 -mypwd xxx -group vgusers -o visgen
dcecp> user create vgadm1 -password vgadm1 -mypwd xxx -group vgusradm -o visgen
dcecp> user create vgadm2 -password vgadm2 -mypwd xxx -group vgusradm -o visgen
dcecp> user create vgos2sj1 -password vgos2sj1 -mypwd xxx -group vgservers -o visgen
dcecp> user create vgos2sj2 -password vgos2sj2 -mypwd xxx -group vgservers -o visgen
dcecp> user create vgwnts1 -password vgwnts1 -mypwd xxx -group vgservers -o visgen
dcecp> user create vgwnts2 -password vgwnts2 -mypwd xxx -group vgservers -o visgen
dcecp> user create vgaixsj1 -password vgaixsj1 -mypwd xxx -group vgservers -o visgen
dcecp> user create vgaixsj2 -password vgaixsj2 -mypwd xxx -group vgservers -o visgen

```

Figure 34. Defining DCE Organizations, Groups, and Users on AIX Cell Server with dcecp. The cell_admin password is used as part of the dcecp command when creating users. Passwords for the individual users are also defined. The -inprojlist yes parameter ensures that users in a group accrue the access rights permitted to the group.

2. Create base directories for DCE-based client/server communication.

VisualAge Generator DCE-based client/server communication support demands a specific base directory structure. This base directory structure is a prerequisite. This base directory structure must be created manually and named `././Servers/VAGenerator`.

You can use the dcecp tool or SMIT to create the base directory structure.

Figure 35 shows the dcecp commands issued (after entering dcecp on the AIX command line) to create the base DCE directory structure required for VisualAge Generator DCE-based client/server communication.

```

dcecp> directory create ././Servers
dcecp> directory create ././Servers/VAGenerator
dcecp> exit

```

Figure 35. Defining Base DCE Directories on AIX Cell Server with dcecp

Target directories specific to the particular application system are created for the configured SERVERID value in this directory structure after authorities for the base directory are defined.

3. Configure the required application server principal authority for the CDS base directory structure.

A DCE principal is identified in the configuration file passed to the DCE server program CSODCES. The DCE principal is used to enter the DCE cell and create or update the objects in the target directory (`././Servers/VAGenerator/SERVERID`).¹⁴ To perform these operations certain privileges must be defined for the principal.

In DCE, authorization for specific tasks is controlled by access control lists (ACLs). Each ACL entry for a user, group, or other DCE resource has a set of allowed authorities. These are the authorities for CDS ACL entries:

¹⁴ During our testing, we determined that the target directory name, as determined by the SERVERID value, was restricted to eight characters. This limit may be removed at a later date. If you require names longer than eight characters make a request to the IBM VisualAge Generator development lab for service.

Token	Description
r	Read entry attributes
w	Update entry attributes
d	Delete entry
t	Test attribute values
c	Change ACL
i	Create new directory entries
a	Administer directory replication

There are multiple types of ACLs for CDS directories:

OBJECT ACL

Controls access to any CDS name (object entries, soft links, child pointers, clearinghouses, and directories).

INITIAL OBJECT Creation ACL

Applies only to CDS directory names. This ACL is assigned to any objects (soft links, application-defined object entries, child pointers, and clearinghouse object entries) to be created in the directory in the future.

INITIAL CONTAINER Creation ACL

Applies only to CDS directory names. This ACL is assigned to any directories to be created in the directory in the future.

If we look at the OBJECT ACL for the base directory (see Figure 36) we can see that the group `subsys/dce/cds-admin` has the required authority.

```
acl_edit /./Servers/VAGenerator -l

# SEC_ACL for /./Servers/VAGenerator:
# Default cell = /.../vgrisc
unauthenticated:r--t---
user:cell_admin:rwdtcia
user:hosts/vgrisc/cds-server:rwdtcia
group:subsys/dce/cds-admin:rwdtcia
group:subsys/dce/cds-server:rwdtcia
any_other:r--t---
```

Figure 36. OBJECT ACL for VisualAge Generator DCE Base Directory

The easiest (but *not* the recommended) way to obtain the CDS privileges required is to add the application server principal to the CDS administrator group (`subsys/dce/cds-admin`). If you are logged in as *cell_admin*, you can issue the `dcecp` command shown in Figure 37 to add user `vgos2sj1` to the CDS administrator group (`cds-admin`).

```
dcecp> group add subsys/dce/cds-admin -member vgos2sj1
```

Figure 37. Adding Application Server Principal to CDS Administrator Group

The problem is that CDS administrator privileges far exceed the actual requirements of an application server principal. The DCE server program `CSODCES` must be able to create and update the objects in the target directory (`/./Servers/VAGenerator/SERVERID`).

However, the group `subsys/dce/cds-admin` has the listed authority (`rwdtcia`) on every directory in the DCE Cell (this is not shown in Figure 36).

We can provide the application server principal with the required authority on only the base directory (and any future target directories) by adding an entry for the

application server principals group to the base directory OBJECT, INITIAL OBJECT, and INITIAL CONTAINER ACLs (see Figure 38 on page 63). With this technique, the ACL authority for the base directory will apply to the target directories (and objects) created in the base directory.

```
acl_edit /./Servers/VAGenerator -m group:vgservers:rwdti
acl_edit /./Servers/VAGenerator -ic -m group:vgservers:rwdti
acl_edit /./Servers/VAGenerator -io -m group:vgservers:rwdti

acl_edit /./Servers/VAGenerator -l

# SEC_ACL for /./Servers/VAGenerator:
# Default cell = /.../vgrisc
unauthenticated:r--t---
user:cell_admin:rwdtcia
user:hosts/vgrisc/cds-server:rwdtcia
group:subsys/dce/cds-admin:rwdtcia
group:subsys/dce/cds-server:rwdtcia
group:vgservers:rwdt-i-
any_other:r--t---
```

Figure 38. Adding Authority for Base Directory to Application Server Group.

Commands to add OBJECT, INITIAL OBJECT, and INITIAL CONTAINER ACLs are shown along with a command that lists the current OBJECT ACL entries for the directory. Similar results would be seen for an INITIAL OBJECT or an INITIAL CONTAINER ACL list.

This prepares our base directory for unauthenticated DCE-based client/server communication.

If you are planning on implementing authenticated DCE-based client/server communication (REMOTE_COMTYPE=dcesecure) you may wish to modify the unauthenticated and any other (any_other) ACL entries of type OBJECT, INITIAL OBJECT, and INITIAL CONTAINER before continuing.

Authenticated DCE-based client/server communication based on VisualAge Generator uses the test ACL authority to determine whether a user is allowed to use a service. The current CDS ACL entries (see Figure 38) provide the test ACL authority to unauthenticated and any other users. In other words, no security is yet available.

The test authority can be removed from the base directory unauthenticated and any other ACL entries with the commands shown in Figure 39 on page 64.

```

acl_edit ./Servers/VAGenerator -m unauthenticated:r-----
acl_edit ./Servers/VAGenerator -io -m unauthenticated:r-----
acl_edit ./Servers/VAGenerator -ic -m unauthenticated:r-----

acl_edit ./Servers/VAGenerator -m any_other:r-----
acl_edit ./Servers/VAGenerator -io -m any_other:r-----
acl_edit ./Servers/VAGenerator -ic -m any_other:r-----

acl_edit ./Servers/VAGenerator -ic -l

# Initial SEC_ACL for directories created under: ./Servers/VAGenerator:
# Default cell = ../../vgrisc
unauthenticated:r-----
group:subsys/dce/cds-admin:rwdtcia
group:subsys/dce/cds-server:rwdtcia
group:vgservers:rwdt-i-
any_other:r-----

```

Figure 39. Removing Test Authority on Base Directory for unauthenticated and any_other ACL Entries. Commands to modify the OBJECT, INITIAL OBJECT, and INITIAL CONTAINER unauthenticated and any_other ACL entries are shown, along with a command that lists the current INITIAL CONTAINER ACL entries for the directory. Similar results would be seen for an OBJECT or an INITIAL OBJECT ACL list.

The changes shown in Figure 38 on page 63 and Figure 39 prepare the base directory for unauthenticated and authenticated VisualAge Generator DCE-based client/server communication without allowing the application server principals unneeded authorization for the full CDS directory structure.

4. Create target directory for application information.

When the CSODCES application is started at the application server, it sends the information contained in the configuration file to the CDS Server. This file contains the values for SERVERID and LOCATION parameters that identify the DCE server program running on an application server workstation that can respond to requests for VisualAge Generator server applications. The SERVERID and LOCATION parameter values are also used in the VisualAge Generator linkage table.

As described in *Developing VisualAge Generator Client/Server Applications*,

SERVERID	Represents an application system or subsystem.
LOCATION	Represents one or more application server workstations that process requests for the application system or subsystem.

In our configuration, we used a SERVERID value of *vgcs*. The location value depended on the active DCE-based client/server communication scenario.¹⁵

CSODCES will write the information about the location and available server applications in a target directory with the same name as the SERVERID value. This directory must exist in a predefined (and hardcoded) base directory named *./Servers/VAGenerator*.

CDS objects with names equal to the LOCATION value and the names of the available VisualAge Generator server applications will be created in the target directory (*./Servers/VAGenerator/SERVERID*).

¹⁵ In the manual *Developing VisualAge Generator Client/Server Applications* DCE examples used a SERVERID value of *TEST*. We confused this with the ACL test authority, which is why we did not use a SERVERID value of *TEST*.

You can use the dcecp tool or SMIT to create the directory. Figure 40 on page 65 shows the dcecp commands issued (after entering dcecp on the AIX command line) to create the target DCE directory required for our VisualAge Generator DCE-based client/server communication scenario.

```
dcecp> directory create ../Servers/VAGenerator/vgcs
```

Figure 40. Creating Target DCE Directory on AIX Cell Server with dcecp

Because we had already implemented our ACL entries for the base directory (see Figure 38 on page 63 and Figure 39 on page 64) we automatically get the authority required on the target directory (see Figure 41).

```
acl_edit ../Servers/VAGenerator/vgcs -l

# SEC_ACL for ../Servers/VAGenerator/vgcs:
# Default cell = ../vgrisc
unauthenticated:r-----
user:cell_admin:rwdtcia
user:hosts/vgrisc/cds-server:rwdtcia
group:subsys/dce/cds-admin:rwdtcia
group:subsys/dce/cds-server:rwdtcia
group:vgservers:rwdt-i-
any_other:r-----
```

Figure 41. OBJECT ACL List for VisualAge Generator DCE Target Directory

Our DCE cell is now ready to support VisualAge Generator DCE-based client/server communication.

4.2.3 OS/2 DCE Application Server Workstation

The OS/2 application server workstation must provide support for DCE client/server communications and for execution of VisualAge Generator server applications.

We used the following software on our OS/2 DCE application server:

- OS/2 Warp Version 3.0 (at Fixpack 21 or later)
- DB2/2 Version 2.1 (for VisualAge Generator server application database support)
- VisualAge C++ for OS/2
- VisualAge Generator Workgroup Services for OS/2 Version 2.2 (Fixpak 3)
- IBM DSS for OS/2 Warp Version 4.0

To implement the DCE application server workstation we had to install and configure DCE and then set up the DCE server program CSODCES. We also generated the OS/2 server portion of the sample application.

4.2.3.1 DCE Client Configuration

DCE client software is required on all workstations in a DCE-based client/server communication environment. Because this workstation (OS/2 DCE application server workstation) will act as DCE application server, we must also install the CDS client component of the DCE client software package.

To configure the DCE client, you must identify the name of your DCE cell and the host names of your cell directory server and the security server. In our configuration, we ran the CDS, security, and DTS servers on the AIX DCE server workstation. On the DCE application server workstation, we configured CDS, security, and DTS clients. We

provided these values and made these decisions when configuring DSS for OS/2 Warp:

Host setup	Cell name: /.../vgrisc DCE host name: vgrisc.raleigh.ibm.com LAN Profile: lan-profile
Protocols	Connection-oriented: yes (checked) Connectionless: yes (checked)
Startup options	Start DCE at system startup: no (not checked) Mode to start components: Single window Synchronize the host's clock before configuring DCE and at DCE startup: yes (checked). Provided a DTS server ID using TCP/IP identification with vgrisc.raleigh.ibm.com as the TCP/IP host name.
Security client	Provided a master security server ID using TCP/IP identification with vgrisc.raleigh.ibm.com as the TCP/IP host name.
Directory client	Provided a directory server ID using TCP/IP identification with vgrisc.raleigh.ibm.com as the TCP/IP host name.
CDS server	Cell Administrator and Password Chose the installation option that supports full configuration of the client. When you select this option, the base configuration tasks for the DCE client are performed automatically on both the DCE client workstation and the remote DCE cell server platform. We used the cell_admin ID and provided the current password.

During our experimentation we disabled our DCE client on OS/2 several times. When this occurred, we would reconfigure to correct the DCE problems. Reconfiguration using the *Configure DCE Services* function required these steps:

1. Select **None** as the services-to-configure option.
2. On the **Cell Administrator and Password** page select **Complete local host configuration** and run the configuration process.
Repeat the previous two steps if any errors occur during the process of removing the configuration of directory, security, or DTS client support.
3. Reconfigure using the same process as the initial configuration activity once you have successfully removed the configuration from all clients.

This would reset our DCE client configuration and allow us to use DCE services again.

Note: Removing and reestablishing the DCE configuration removed any entries we had made in the keytab file on the DCE application server workstation. We had to use rgy_edit to add the principal again (see Figure 54 on page 77).

4.2.3.2 Build VisualAge Generator Server Applications

Three sample application servers were generated for use in the DCE configuration. Two support remote calls from VisualAge Generator clients using DCE-based client/server communication support, the third is called as a local server. The servers are described as follows:

- We have two servers that are part of the sample application that must be generated for OS/2 with DCE-based client/server communication support: VGD2OS2 and VGD3OS2. Server VGD2OS2 is called by the sample application GUI client. Server VGD3OS2 is called by a sample application server when a three-tier server call to an OS/2 platform using DCE-based client/server communication is requested.

- Server VGL3OS2 is generated with a local call interface. VGL3OS2 can be called by VGD2OS2 to implement an alternative third-tier call.

Two further steps need to be carried out:

1. Define linkage table for DCE sample application servers. Figure 42 shows the linkage table entries we used to generate the VGD2OS2, VGD3OS2, and VGL3OS2 applications.

```
:CALLLINK APPLNAME=VGD2OS2 LIBRARY=VGD2OS2 REMOTECOMTYPE=dce
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgos2 REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD3OS2 LIBRARY=VGD3OS2 REMOTECOMTYPE=dce
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgos2 REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGL3OS2 LIBRARY=VGL3OS2 PARMFORM=osl link
          LINKTYPE=dynamic LUWCONTROL=server REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD3WNT LIBRARY=VGD3WNT REMOTECOMTYPE=dce
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgwnt REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD3AIX LIBRARY=VGD3AIX REMOTECOMTYPE=dce
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgaix REMOTEBIND=runtime
          contable=binary.
```

Figure 42. Linkage Table for OS/2 Sample Application Servers with DCE-Based Client/Server Communication Support. Linkage table entries are included for both the servers being generated and the servers that can be called from VGD2OS2.

2. Generate sample application servers for OS/2 with DCE-based client/server communication support.

Figure 43 contains the generation commands we used to generate the sample application servers for OS/2 with DCE-based client/server communication support.

```
EZE2GEN GENERATE VGD2OS2 /MSL=VGPI NGZ /GENOUT=e:\vgcs\genout\%%EZEENV%%
          /linkage=h:\vgcs\applst.1kg /system=OS2
          >e:\vgcs\genout\VGD2OS2.log

EZE2GEN GENERATE VGD3OS2 /MSL=VGPI NGZ /GENOUT=e:\vgcs\genout\%%EZEENV%%
          /linkage=h:\vgcs\applst.1kg /system=OS2
          >e:\vgcs\genout\VGD3OS2.log

EZE2GEN GENERATE VGL3OS2 /MSL=VGPI NGZ /GENOUT=e:\vgcs\genout\%%EZEENV%%
          /linkage=h:\vgcs\applst.1kg /system=OS2
          >e:\vgcs\genout\VGL3OS2.log
```

Figure 43. Generating OS/2 Sample Application Servers with DCE-Based Client/Server Communication Support. Both the local and the remote DCE-based sample application servers are generated for OS/2. This provides support for multiple two- and three-tier computing configurations.

You are now ready to configure and start the DCE server program on this workstation. This process is nearly identical on OS/2, Windows NT, and AIX server platforms. For this reason, we describe it only once in 4.2.7, "Running Applications with DCE-Based Client/Server Communication" on page 75.

4.2.4 OS/2 DCE GUI Client Workstation

The OS/2 client workstation must provide support for DCE client/server communications and execution of VisualAge Generator GUI client applications.

We used the following software for the OS/2 DCE GUI Client:

- OS/2 Warp Version 3.0 (at Fixpack 21 or later)
- TCP/IP support
- VisualAge Generator GUI Runtime for OS/2 Version 2.2 (Fixpak 3)
- IBM DSS for OS/2 Warp Version 4.0

To implement the DCE GUI client workstation, we had to install and configure DCE. We also generated the OS/2 GUI client portion of the sample application.

4.2.4.1 DCE Client Configuration

DCE client software is required on all workstations in a DCE-based client/server communication environment. VisualAge Generator GUI application calls using DCE-based client/server communication use only a subset of the function available in a full DCE client installation. This means that we can install just the slim client component of DCE. The CDS, security, or DTS client functions do not need to be installed on the client workstation as they were on the application server workstation (see 4.2.3, "OS/2 DCE Application Server Workstation" on page 65 or 4.2.5, "Windows NT DCE Application Server Workstation" on page 70). The slim client software is enough to query the cell server and send RPC calls to the application server workstation.

To configure the DCE slim client you need the cell name and the TCP/IP names or addresses of the CDS and security server. In our configuration, we ran the CDS, security, and DTS servers on the AIX DCE server workstation. You do not have to provide the cell administrator's password, because configuration tasks are not performed on the DCE cell server. We provided these values and made these decisions when configuring the DCE slim client using DSS for OS/2 Warp:

Host setup	Cell name: /.../vgrisc DCE host name: vgrisc.raleigh.ibm.com LAN Profile: lan-profile
Protocols	Connection-oriented: yes (checked) Connectionless: yes (checked)
Startup options	Start DCE at system startup: no (not checked) Mode to start components: Single window Synchronize the host's clock before configuring DCE and at DCE startup: yes (checked). Provides a DTS server ID using TCP/IP identification with vgrisc.raleigh.ibm.com as the TCP/IP host name.
Slim client	Provides a master security server ID using TCP/IP identification with vgrisc.raleigh.ibm.com as the TCP/IP host name. Provides a directory server ID using TCP/IP identification with vgrisc.raleigh.ibm.com as the TCP/IP host name.

During our experimentation, we disabled our DCE client on OS/2 several times. When this occurred, we would reconfigure to correct the DCE problems. Reconfiguration using the *Configure DCE Services* function required these steps:

1. Select **None** as the services to configure option.
2. On the *Cell Administrator and Password* page select **Complete local host configuration** and run the configuration process.

Repeat both steps if any errors occur during the configuration removal from directory, security, or DTS client support.

3. Reconfigure using the same process as the initial configuration activity once you have successfully removed the configuration of all clients.

This would reset our DCE client configuration and allow us to use DCE services again.

4.2.4.2 Build VisualAge Generator Client Platform Applications

The OS/2 client platform portion of the sample application had to be generated with support for remote calls to VisualAge Generator servers using DCE-based client/server communication support.

The sample application GUI client can directly call a second-tier DCE-based server application on the OS/2, Windows NT, or AIX platforms. Third-tier DCE-based servers could also be called from the OS/2 client platform if we generated the local C++ server included in the sample application (VGL2OS2).

The following steps are required to build the VisualAge Generator client platform applications:

1. Define linkage table for all possible OS/2 client platform calls to DCE sample application servers.

Figure 44 shows the linkage table entries we used to generate the sample application GUI client and local C++ server to support second- or third-tier calls to OS/2, Windows NT, or AIX platforms.

```
:CALLLINK APPLNAME=VGL2OS2 LIBRARY=VGL2OS2  PARMFORM=oslink
          LINKTYPE=dynamic LUWCONTROL=server REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD2OS2 LIBRARY=VGD2OS2  REMOTECOMTYPE=dce
          PARMFORM=oslink  LINKTYPE=remote  LUWCONTROL=server
          SERVERID=vgcs    LOCATION=vgos2   REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD2WNT LIBRARY=VGD2WNT  REMOTECOMTYPE=dce
          PARMFORM=oslink  LINKTYPE=remote  LUWCONTROL=server
          SERVERID=vgcs    LOCATION=vgwnt   REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD2AIX LIBRARY=VGD2AIX  REMOTECOMTYPE=dce
          PARMFORM=oslink  LINKTYPE=remote  LUWCONTROL=server
          SERVERID=vgcs    LOCATION=vgaix   REMOTEBIND=runtime
          contable=binary.
```

Figure 44. Linkage Table for OS/2 Sample Application Client with DCE-Based Client/Server Communication Support

Note: When using DCE-based client/server communication, the LUW is always controlled by the server. It is not possible to have client LUW control using DCE-based client/server communication.

2. Generate a sample application client for OS/2 with DCE-based client/server communication support.

Figure 45 on page 70 contains the generation commands we used to generate the sample application client for OS/2 with DCE-based client/server communication support.

```

EZE2GEN GENERATE VGPING /MSL=VGPINGZ
/GENOUT=e:-VGCS-GENOUT-OS2 /linkage=h:-vgcs-applst.lkg
/system=OS2gui >e:-vgcs-genout-VGPING.log

EZE2GEN GENERATE VGPINGSD /MSL=VGPINGZ
/GENOUT=e:-VGCS-GENOUT-OS2 /linkage=h:-vgcs-applst.lkg
/system=OS2gui >e:-vgcs-genout-VGPINGsd.log

EZE2GEN GENERATE VGHELP /MSL=VGPINGZ
/GENOUT=e:-VGCS-GENOUT-OS2 /linkage=h:-vgcs-applst.lkg
/system=OS2gui >e:-vgcs-genout-vghelp.log

EZE2GEN GENERATE VGL2OS2 /MSL=VGPINGZ
/GENOUT=e:-VGCS-GENOUT-OS2 /linkage=h:-vgcs-applst.lkg
/system=OS2 >e:-vgcs-genout-VGL2OS2.log

```

Figure 45. Sample Application Generation Commands for OS/2 Client for DCE-Based Client/Server Communication. Embedded GUI applications are generated because of the /GENEMBEDDEDGUI default generation option.

4.2.5 Windows NT DCE Application Server Workstation

The Windows NT application server workstation must provide support for DCE client/server communications and for execution of VisualAge Generator server applications.

We used the following software on our Windows NT DCE application server:

- Windows NT V3.51 with TCP/IP support
- DB2/NT Version 2.1 with SDK support (for VisualAge Generator server application database support)
- VisualAge C++ for Windows NT
- VisualAge Generator Workgroup Services for Windows NT Version 2.2 (Fixpak 3)
- Gradient PC-DCE/32 V 1.0.3a

To implement the DCE application server workstation, we had to install and configure DCE and then set up the DCE server program CSODCES. We also generated the Windows NT server portion of the sample application. To install DCE support on Windows NT we did the following:

1. Installed Gradient PC-DCE/32 V 1.0.3a with run-time kit option.
2. Installed run-time license pack for Windows NT.

4.2.5.1 DCE Client Configuration

DCE client software is required on all workstations involved in a DCE-based client/server communication environment.

The DCE server program CSODCES provided by VisualAge Generator requires a full DCE client installation. This means that we must install or configure support for the CDS Client as part of the DCE installation.

To configure full DCE client support with Gradient's PC-DCE/32, you need the cell name and the TCP/IP names or addresses of the CDS and Security server. In our configuration, we ran the CDS, security, and DTS servers on the AIX DCE server workstation. We provided these values and made the following decisions when using the PC-DEC/32 configuration notebook for Windows NT.

Note: The PC-DEC/32 configuration notebook is displayed when the **Configure** push button is clicked on the DCE Service Panel. The DCE Service Panel can be started using the PC-DCE/32 icon in the Windows NT Control Panel.

On the *DCE Client* page:

Cell name:	/.../vgrisc
Server information	Security: vgrisc.raleigh.ibm.com as the TCP/IP host name. Cell Directory: vgrisc.raleigh.ibm.com as the TCP/IP host name.
Configure client daemons	Checked When you request (check) the Configure Client Daemons option you do not have to provide the cell administrator's principal ID and password because configuration tasks are performed on the DCE cell server.
Start rpcd	Unchecked This is grayed out (and ignored) when the Configure Client Daemons option is checked. No other options were configured on this page.

On the *Options* page none of the available check boxes were selected.

4.2.5.2 Build VisualAge Generator Server Applications

Three sample application servers were generated for use in the DCE configuration. Two support remote calls from VisualAge Generator clients using DCE-based client/server communication support; the third is called as a local server. The servers are described as follows:

- We have two servers that are part of the sample application that must be generated for Windows NT with DCE-based client/server communication support: VGD2WNT and VGD3WNT. Server VGD2OS2 is called by the sample application GUI client. Server VGD3OS2 is called by a sample application server when a three-tier server call to an Windows NT platform using DCE-based client/server communication is requested.
- Server VGL3WNT is generated with a local call interface. VGL3WNT can be called by VGD2WNT to implement an alternative third tier call.

The following steps are required to build the VisualAge Generator server applications:

1. Define linkage table for DCE sample application servers. Figure 46 on page 72 shows the linkage table entries we used to generate the VGD2WNT, VGD3WNT, and VGL3WNT applications.

```

:CALLLINK APPLNAME=VGD2WNT LIBRARY=VGD2WNT REMOTECOMTYPE=dce
          PARMFORM=oslink LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgwnt REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD3WNT LIBRARY=VGD3WNT REMOTECOMTYPE=dce
          PARMFORM=oslink LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgwnt REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGL3WNT LIBRARY=VGL3WNT PARMFORM=oslink
          LINKTYPE=dynamic LUWCONTROL=server REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD3OS2 LIBRARY=VGD3OS2 REMOTECOMTYPE=dce
          PARMFORM=oslink LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgwnt REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD3AIX LIBRARY=VGD3AIX REMOTECOMTYPE=dce
          PARMFORM=oslink LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgaix REMOTEBIND=runtime
          contable=binary.

```

Figure 46. Linkage Table for Windows NT Sample Application Servers with DCE-Based Client/Server Communication Support. Linkage table entries for both the servers being generated and servers that can be called from VGD2WNT are included.

2. Generate sample application servers for Windows NT with DCE-based client/server communication support.

Figure 47 contains the generation commands we used to generate the sample application servers for OS/2 with DCE-based client/server communication support.

```

EZE2GEN GENERATE VGD2WNT /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%%EZEENV%%
               /linkage=h:\vgcs\applst.lkg /system=WINNT
               >e:\vgcs\genout\VGD2WNT.log

EZE2GEN GENERATE VGD3WNT /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%%EZEENV%%
               /linkage=h:\vgcs\applst.lkg /system=WINNT
               >e:\vgcs\genout\VGD3WNT.log

EZE2GEN GENERATE VGL3WNT /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%%EZEENV%%
               /linkage=h:\vgcs\applst.lkg /system=WNNT
               >e:\vgcs\genout\VGL3WNT.log

```

Figure 47. Generating Windows NT Sample Application Servers with DCE-Based Client/Server Communication Support. Both the local and the remote DCE-based sample application servers are generated for Windows NT. This provides support for multiple two- and three-tier computing configurations.

You are now ready to configure and start the DCE server program on this workstation. This process is nearly identical on OS/2, Windows NT, and AIX server platforms. For this reason, we describe it only once in 4.2.7, "Running Applications with DCE-Based Client/Server Communication" on page 75.

4.2.6 Windows 95 DCE GUI Client Workstation

The Windows 95 client workstation must provide support for DCE client/server communications and execution of VisualAge Generator GUI client applications.

We used the following software for the Windows 95 DCE GUI Client:

- Windows 95
- TCP/IP (as provided by Windows 95)
- VisualAge Generator GUI Run Time for Windows Version 2.2 (Fixpak 3)
- Gradient Technologies PC-DCE/32 for Windows 95

To implement the DCE GUI client workstation we had to install and configure DCE. We also generated the Windows 95 GUI client portion of the sample application. To install DCE support on Windows 95, we did the following:

1. Installed Gradient PC-DCE/32 V 1.0.3a with run-time kit option
2. Installed run-time license pack for Windows 95

4.2.6.1 DCE Client Configuration

DCE client software is required on all workstations involved in a DCE-based client/server communication environment.

VisualAge Generator GUI application calls using DCE-based client/server communication use only a subset of the function available in a full DCE client installation. This means that we can install or configure just the RPC function as provided by PC-DCE/32. The the CDS, security, or DTS client functions do not need to be configured on the client workstation as they were on the application server workstation (see 4.2.3, "OS/2 DCE Application Server Workstation" on page 65 or 4.2.5, "Windows NT DCE Application Server Workstation" on page 70). The base DCE configuration is enough to query the the cell server and send RPC calls to the application server workstation.

To configure RPC support with Gradient's PC-DCE/32, you need the cell name and the TCP/IP names or addresses of the CDS and Security server. In our configuration, we ran the CDS, security, and DTS servers on the AIX DCE server workstation. We provided these values and made the following decisions when using the PC-DEC/32 configuration notebook for Windows 95.

Note: The PC-DEC/32 configuration notebook is displayed when the **Configure** push button is clicked on the DCE Service Panel. The DCE Service Panel can be started using the PC-DCE/32 icon in the Windows 95 Control Panel.

On the *DCE Client* page:

Cell name:	/.../vgrisc
Server information	Security: vgrisc.raleigh.ibm.com as the TCP/IP host name. Cell Directory: vgrisc.raleigh.ibm.com as the TCP/IP host name.
Configure client daemons	Unchecked If you do not request (check) the Configure Client Daemons option, you do not have to provide the cell administrator's principal ID or password because configuration tasks are not performed on the DCE cell server.
Start rpcd	Checked No other options were configured on this page.

On the *Options* page, none of the available check boxes were selected.

4.2.6.2 Build VisualAge Generator Client Applications

The client portion of the sample application had to be generated with support for remote calls to VisualAge Generator servers using DCE-based client/server communication support.

The sample application client can directly call on DCE-based server application: VGD2OS2.

The following steps are required to build the VisualAge Generator client applications:

1. Define linkage table for all possible Windows 95 client platform calls to DCE sample application servers. Figure 48 shows the linkage table entries we used to generate the sample application GUI client and local C++ server to support second- or third-tier calls to OS/2, Windows NT, or AIX platforms.

```
:CALLLINK APPLNAME=VGD2OS2 LIBRARY=VGD2OS2 REMOTECOMTYPE=dce
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgos2 REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD2WNT LIBRARY=VGD2WNT REMOTECOMTYPE=dce
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgwnt REMOTEBIND=runtime.

:CALLLINK APPLNAME=VGD2AIX LIBRARY=VGD2AIX REMOTECOMTYPE=dce
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgaix REMOTEBIND=runtime
          contable=binary.
```

Figure 48. Linkage Table for Windows 95 Sample Application Client with DCE-Based Client/Server Communication Support

Note: When using DCE-based client/server communication, the LUW is always controlled by the server. It is not possible to have client LUW control using DCE-based client/server communication.

2. Generate sample application client for Windows 95 with DCE-based client/server communication support

Figure 49 contains the generation commands we used to generate the sample application client for Windows 95 with DCE-based client/server communication support.

```
EZE2GEN GENERATE VGPI NG /MSL=VGPI NGZ
          /GENOUT=e:-VGCS-GENOUT-wingui /linkage=h:-vgcs-applst.lkg
          /system=wingui >e:-vgcs-genout-VGPI NG.logNT

EZE2GEN GENERATE VGPI NGSD /MSL=VGPI NGZ
          /GENOUT=e:-VGCS-GENOUT-wingui /linkage=h:-vgcs-applst.lkg
          /system=wingui >e:-vgcs-genout-VGPI NGsd.logNT

EZE2GEN GENERATE VGHELP /MSL=VGPI NGZ
          /GENOUT=e:-VGCS-GENOUT-wingui /linkage=h:-vgcs-applst.lkg
          /system=wingui >e:-vgcs-genout-VGHELP.logNT
```

Figure 49. Sample Application Generation Commands for Windows 95 GUI Client for DCE-Based Client/Server Communication. Generation of embedded GUI applications will occur because of the /GENEMBEDDEDGUI default generation option.

4.2.7 Running Applications with DCE-Based Client/Server Communication

Now that the environment setup is complete (DCE Cell configured, DCE software installed on client and server platforms, applications generated), we are ready to run the sample application using DCE-based client/server communication.

We implement the DCE-based client/server communication configuration shown in Figure 50.

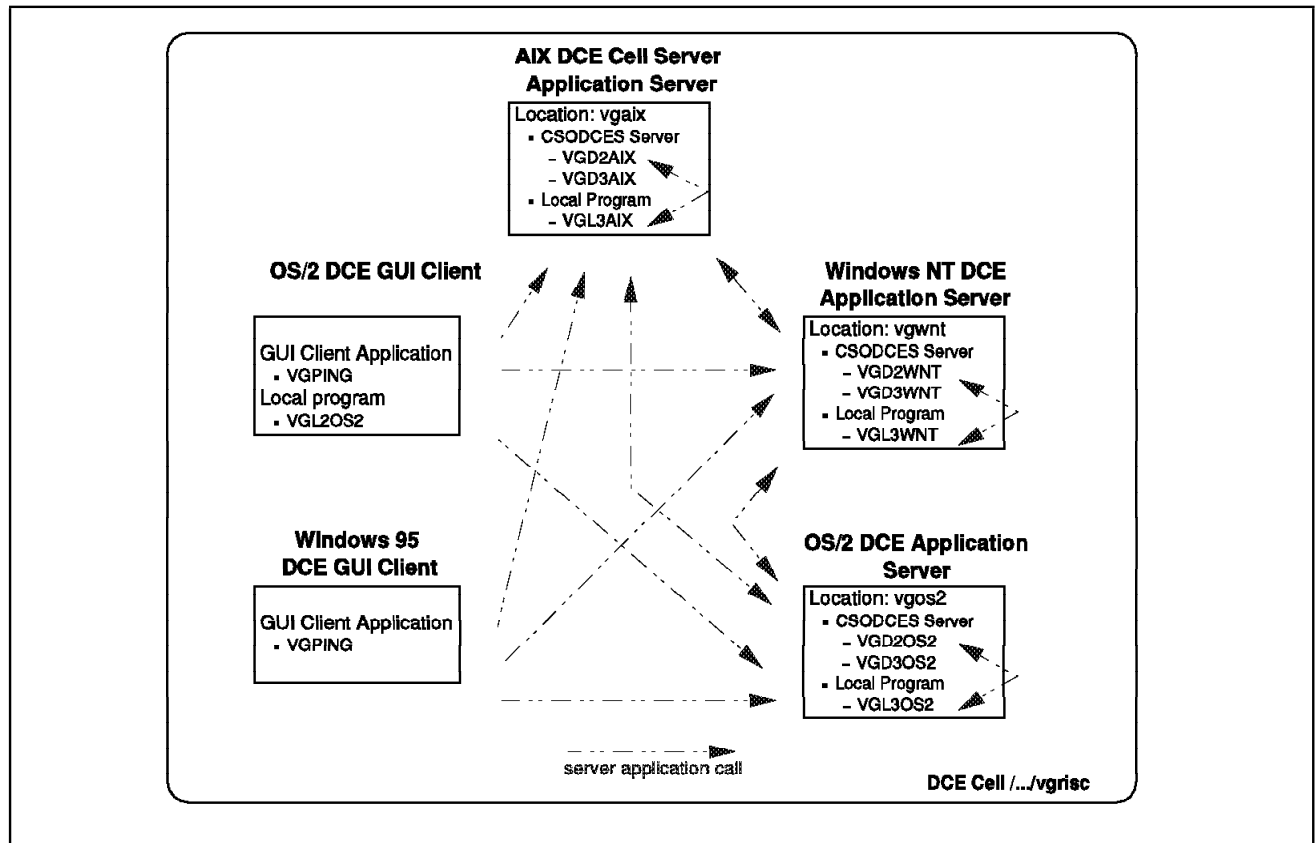


Figure 50. VisualAge Generator DCE-Based Client and Application Server Configuration

Implementation requires that we first configure and then start the DCE server program (CSODCES), learn how it interacts with the directory setup in the DCE cell, and then start the client end of the sample application. We also study techniques for implementing multiple application server workstation platforms for one SERVERID/LOCATION pair.

4.2.7.1 DCE Server Program Configuration

Several steps are required to configure the CSODCES DCE server program:

1. Define CSODCES configuration file

The CSODCES DCE server program requires a configuration file as a parameter for startup. Figure 51 on page 76 contains the DCE.CNF configuration file we used on our OS/2 DCE application server workstation.

```
DCEprincipal=vgos2sj1
LOCATION=vgos2
SERVERID=vgcs
DCEACLobject=./Servers/VAGenerator/vgcs/vgos2
SECURE APPLICATIONS=
PUBLIC APPLICATIONS=
VGD2OS2
VGD3OS2
```

Figure 51. DCE.CNF Configuration File for OS/2 DCE Application Server

This configuration file defines how the CSODCES program will operate. The following parameters are defined:

DCEprincipal DCE principal that will be used for CDS access. This principal is also defined in the DCE keytab file using `rgy_edit`.

LOCATION Name of the workstation where the VisualAge Generator server applications are located. This might represent a set of workstations if the DCE server program is started on more than one workstation with the same `LOCATION` value.

SERVERID Name for the system or subsystem

DCEACLobject Name of the active access control list (ACL) object used for secure communication (`REMOTECOMTYPE=dcesecure`)

A list of available VisualAge Generator server applications is also included in the CSODCES program configuration file. These VisualAge Generator server applications are defined as either:

PUBLIC No authorization checking is done by CSODCES for client requests for these applications. A DCE logon is not required on the client platform.

SECURE CSODCES verifies that the DCE client requesting these applications is authorized to access them. A DCE logon is required on the client platform if the appropriate ACL definitions have been made (see Figure 39 on page 64). The implementation of SECURE processing is further explained in 4.2.8, "Implementing Secure DCE-Based Client/Server Communication" on page 88.

The `SERVERID` and `LOCATION` values used in the CSODCES configuration file must match the `SERVERID` and `LOCATION` values used in the linkage table referenced at run time.

Figure 52 contains the DCE.CNF configuration file we used on our Windows NT DCE application server workstation.

```
DCEprincipal=vgwntsj1
LOCATION=vgwnt
SERVERID=vgcs
DCEACLobject=./Servers/VAGenerator/vgcs/vgwnt
SECURE APPLICATIONS=
PUBLIC APPLICATIONS=
VGD2WNT
VGD3WNT
```

Figure 52. DCE.CNF Configuration File for Windows NT DCE Application Server

Figure 53 on page 77 contains the DCE.CNF configuration file we used on our AIX DCE application server workstation.

```
DCEprincipal=vgaixsj1
LOCATION=vgaix
SERVERID=vgcs
DCEACLobject=../../Servers/VAGenerator/vgcs/vgaix
SECURE APPLICATIONS=
PUBLIC APPLICATIONS=
VGD2AIX
VGD3AIX
```

Figure 53. DCE.CNF Configuration File for AIX DCE Application Server

2. Add a key table entry for the DCE principal

The CSODCES DCE server program uses a dynamic DCE logon to access and update the target directory with information about the active location and available VisualAge Generator server applications. The principal used during the logon is defined in the configuration file but the password is stored in a local DCE keytab file.

Before starting the CSODCES DCE server program, you have to make an entry in the keytab file on the DCE application server workstation for the DCE principal identified in the CSODCES configuration file. Use the rgy_edit command on each DCE application server workstation to define the DCE principal used by the CSODCES DCE server program. Figure 54 shows the rgy_edit definition process we used on our OS/2 DCE application server workstation.

```
[E:\vgcs\dcesec]rgy_edit
Current site is: registry server at ../../vgrisc
rgy_edit=> do princ
Domain changed to: principal
rgy_edit=> ktadd -p vgos2sj1 -pw vgos2sj1
rgy_edit=> ktlist
../../vgrisc/hosts/vgrisc.almaden.ibm.com/self      1
../../vgrisc/hosts/vgrisc.almaden.ibm.com/self      2
../../vgrisc/vgos2sj1                               1
rgy_edit=> exit
bye.

[E:\vgcs\dcesec]
```

Figure 54. Adding Key Table Entry for DCE Application Server Principal

You need to add the principal to the keytab on each DCE application server workstation you configure. The rgy_edit interface is the same regardless of operating system environment (OS/2, Windows NT, AIX). If you reconfigure DCE software, the keytab file is often reinitialized. This may require that you add the principal again.

4.2.7.2 Start and Manage DCE Server Program

There are three steps to follow:

1. Start DCE (if not automatically started).

We configured DCE on our OS/2 and Windows workstations so that we managed startup and shutdown. Before we can run the DCE server program CSODCES, we must ensure that DCE has been started:

- To start DCE on OS/2, use the *Start DCE* program icon in the DCE Services folder. The DCE task window provides details on the services that have started.
- To start DCE on Windows 95 or Windows NT click on the **Start DCE** push button on the DCE Service Panel. The DCE Service Panel can be opened using the PC-DCE/32 icon in the Windows 95 or Windows NT Control Panel.

Sometimes DCE services will not start. Often, the problem is a time skew between the workstation and the time as known to the DCE Cell. DCE can be configured to synchronize the time on the workstation with the time as known to the DCE Cell, which prevents this time-skew problem.

2. Configure runtime support for VisualAge Generator server applications.

VisualAge Generator Workgroup Services must be installed and configured. The VisualAge Generator server applications must be in a directory that is included in the LIBPATH for the operating environment.

Database support is configured using the EZERSQLDB environment variable just as with stand-alone or local VisualAge Generator applications. If the VisualAge Generator server applications called with DCE-based client/server communication is to call other servers, a linkage table should be defined and identified.

Other VisualAge Generator Workgroup Services runtime options, such as trace support, are applicable to DCE-based server applications. We used these settings on our OS/2 workstation during our testing to assist in problem determination:

```
EZERSQLDB=SAMPLE
CSOLINKTBL=H:\VGCS\ACTIVE.LKG
CSOTROPT=2
FCWTROPT=31
```

We used similar settings on the Windows NT and AIX workstations. Tracing communications and application execution in a production environment would be undesirable because of the heavy overhead involved.

3. Start the DCE server program.

When the DCE server program is started, it uses the configuration to determine where to advertise the available VisualAge Generator server applications. Entries are created in the CDS directory identified by the SERVERID value to indicate the location and the available VisualAge Generator server applications.

The CDS directory structure created earlier (see Figure 35 on page 61 and Figure 40 on page 65) is shown in Figure 55.

```
d      ./Servers
d      ./Servers/VAGenerator
d      ./Servers/VAGenerator/vgcs
```

Figure 55. CDS Directories for SERVERID on AIX Cell Server. The **d** signifies that the entry is a directory. (We cut/paste this information from the CDS list shown in smitty.) We are using **vgcs** as the SERVERID value in our DCE configuration file and linkage table.

We started the CSODCES DCE server program on the OS/2 DCE server workstation with this command:

```
csodces -c vgos2.cnf
```

This tells CSODCES to start a primary DCE server program using the configuration file, vgos2.cnf. See 4.2.7.4, "Implementing Multiple DCE Application Server Workstations" on page 84 for a discussion of CSODCES startup parameters for primary and secondary DCE server programs.

Figure 56 on page 79 shows the display for a running DCE server program. The contents of the configuration file are visible in the display.

```
[E:\vgcs\dcesec]csodces vgos2.cnf
DCEprincipal TOKEN=vgos2sj1
LOCATION TOKEN=vgos2
SERVERID TOKEN=vgcs
DCEACLOBJECT TOKEN=../Servers/VAGenerator/vgcs/vgos2
SECURE APPLICATIONS =
PUBLIC APPLICATIONS =
VGD2OS2
VGD3OS2
Server to be loaded VGD2OS2.
Server to be loaded VGD3OS2.
Registering server interface with RPC runtime...
Registering server endpoints with endpoint mapper (RPCD)...
Exporting server bindings into CDS namespace...
Server ../Servers/VAGenerator/vgcs/vgos2 listening...
```

Figure 56. Display for a Running CSODCES DCE Server Program

Figure 57 shows the object entries for the identified location and the available VisualAge Generator server applications that were added to the directory structure shown in Figure 55 on page 78.

```
d      ../Servers
d      ../Servers/VAGenerator
d      ../Servers/VAGenerator/vgcs
o      ../Servers/VAGenerator/vgcs/VGD2OS2
o      ../Servers/VAGenerator/vgcs/VGD3OS2
o      ../Servers/VAGenerator/vgcs/vgos2
```

Figure 57. CDS Directories and Application Objects on AIX Cell Server. The **o** signifies that the entry is an object. We are using **vgos2** as the LOCATION value in our DCE configuration file and linkage table for the OS/2 DCE server workstation. VGD2OS2 and VGD3OS2 are the names of the VisualAge Generator server applications available at this location for the SERVERID system.

The CDS object **vgos2** is used to locate the application server workstation where the requested VisualAge Generator server applications can be found.

We can see this information as stored in the CDS object by DCE using DCE commands (see Figure 58 on page 80).

```

cdscp> show object /./Servers/VAGenerator/vgcs/vgos2

      SHOW
      OBJECT /.../vgrisc/Servers/VAGenerator/vgcs/vgos2
      AT 1997-02-18-19:59:22
RPC_ClassVersion = 0100
RPC_ObjectUUIDs = 80d3f926e489d0119b3708005a94dcc5
RPC_ObjectUUIDs = 602a1728e489d0119b3708005a94dcc5
      CDS_CTS = 1997-02-18-23:10:23.761560100/10-00-5a-b1-f3-f8
      CDS_UTS = 1997-02-19-00:45:07.281280100/10-00-5a-b1-f3-f8
      CDS_Class = RPC_Server
CDS_ClassVersion = 1.0
CDS_Towers = :
      Tower = ncadg_ip_udp:129.33.160.215[]

cdscp>

```

Figure 58. Information for a CDS Object as Shown Using cdscp. The highlighted tower data includes the TCP/IP address for the OS/2 application server workstation.

4.2.7.3 Running the Sample Application Using DCE Communication

Before you can run the GUI application and use DCE-based client/server communication to call servers, you must start the DCE client.

You can log into the DCE cell, using a DCE user ID, if you want. However, this is not required unless you have implemented secure DCE-based client/server communication (REMOTECOMTYPE=dcesecure).

You must also set the CSOLINKTBL environment variable to identify the linkage table file to be used for remote calls. The linkage table entries shown in Figure 44 on page 69 or Figure 48 on page 74 are used to both generate and run the GUI application.

The sample application, as implemented in the DCE configuration shown in Figure 32 on page 58, can call servers on OS/2, Windows NT, or AIX.

Figure 59 shows the object entries for the available DCE application server workstations and VisualAge Generator server applications.

```

d      /./Servers
d      /./Servers/VAGenerator
d      /./Servers/VAGenerator/vgcs
o      /./Servers/VAGenerator/vgcs/VGD2AIX
o      /./Servers/VAGenerator/vgcs/VGD2OS2
o      /./Servers/VAGenerator/vgcs/VGD2WNT
o      /./Servers/VAGenerator/vgcs/VGD3AIX
o      /./Servers/VAGenerator/vgcs/VGD3OS2
o      /./Servers/VAGenerator/vgcs/VGD3WNT
o      /./Servers/VAGenerator/vgcs/vgaix
o      /./Servers/VAGenerator/vgcs/vgos2
o      /./Servers/VAGenerator/vgcs/vgwnt

```

Figure 59. Application Objects for OS/2, Windows NT, and AIX DCE Application Server Workstations

When the sample application GUI client issues a call to a DCE-based server, the location object is used to identify the DCE application server workstation that can satisfy the call (see Figure 58). The call, a DCE remote procedure call request, is then passed using DCE-based client/server communication to the server platform.

When the tracing environment variable has been set (CSOTROPT=2), the process of identifying the DCE application server workstation and making the call is visible in the VisualAge Generator client/server communication trace log (see Figure 60 on page 81 for an extract of the trace log from our Windows 95 client platform).

```

<Feb 18 15:10:46>->CMINIT
<Feb 18 15:10:46><-CMINIT - 0.000
<Feb 18 15:10:46>->CMCALL
<Feb 18 15:10:46> Calling application VGD20S2
<Feb 18 15:10:46> ->readFromLinkTbl
<Feb 18 15:10:46> <-readFromLinkTbl - 0.160
<Feb 18 15:10:46> ->loadAndInitDriver
<Feb 18 15:10:47> <-loadAndInitDriver - 1.040
<Feb 18 15:10:47> ->CMDV_INIT
<Feb 18 15:10:47> <-CMDV_INIT - 0.060
<Feb 18 15:10:47> ->DCE:CMDV_CALL
<Feb 18 15:10:47> ++DCE:CMDV_CALL
<Feb 18 15:10:47> +++DCE:CreateParmBlock
<Feb 18 15:10:47> ====0.000
<Feb 18 15:10:51> Using binding handle: 26f9d380-89e4-11d0-9b37-08005a94dcc5@ncadg_ip_udp:129.33.160.215[]
<Feb 18 15:10:51> Passing 428 bytes of data
<Feb 18 15:10:51> +++DCE:RPCCall
<Feb 18 15:10:56> ====5.060
<Feb 18 15:10:56> ==9.280
<Feb 18 15:10:56> <-DCE:CMDV_CALL
<Feb 18 15:10:57><-CMCALL - 10.710
<Feb 18 15:11:00>->CMCALL
<Feb 18 15:11:00> Calling application VGD20S2
<Feb 18 15:11:00> ->readFromLinkTbl
<Feb 18 15:11:00> <-readFromLinkTbl - 0.000
<Feb 18 15:11:00> ->DCE:CMDV_CALL
<Feb 18 15:11:00> ++DCE:CMDV_CALL
<Feb 18 15:11:00> +++DCE:CreateParmBlock
<Feb 18 15:11:00> ====0.050
<Feb 18 15:11:00> Using binding handle: 26f9d380-89e4-11d0-9b37-08005a94dcc5@ncadg_ip_udp:129.33.160.215[]
<Feb 18 15:11:00> Passing 428 bytes of data
<Feb 18 15:11:00> +++DCE:RPCCall
<Feb 18 15:11:03> ====2.690
<Feb 18 15:11:03> ==2.850
<Feb 18 15:11:03> <-DCE:CMDV_CALL
<Feb 18 15:11:03><-CMCALL - 3.020
<Feb 18 15:11:10>->CMCALL
<Feb 18 15:11:10> Calling application VGD2AIX
<Feb 18 15:11:10> ->readFromLinkTbl
<Feb 18 15:11:10> <-readFromLinkTbl - 0.000
<Feb 18 15:11:10> ->DCE:CMDV_CALL
<Feb 18 15:11:10> ++DCE:CMDV_CALL
<Feb 18 15:11:10> +++DCE:Convert
<Feb 18 15:11:10> ->CMCONV
<Feb 18 15:11:10> <-CMCONV - 0.000
<Feb 18 15:11:10> ->CMCONV
<Feb 18 15:11:10> <-CMCONV - 0.060
<Feb 18 15:11:10> ->CMCONV
<Feb 18 15:11:10> <-CMCONV - 0.050
<Feb 18 15:11:10> ->CMCONV
<Feb 18 15:11:10> <-CMCONV - 0.000
<Feb 18 15:11:10> ====0.220
<Feb 18 15:11:10> +++DCE:CreateParmBlock
<Feb 18 15:11:10> ====0.000
<Feb 18 15:11:12> Using binding handle: f80bd000-89e3-11d0-a4e4-10005ab1f3f8@ncadg_ip_udp:9.67.172.228[]
<Feb 18 15:11:12> Passing 428 bytes of data
<Feb 18 15:11:12> +++DCE:RPCCall
<Feb 18 15:11:15> ====3.740
<Feb 18 15:11:15> ==5.600

```

Figure 60 (Part 1 of 2). VisualAge Generator Windows 95 Client Trace Log for DCE-Based Client/Server Communication Calls

```

<Feb 18 15:11:15> ++DCE:unConvert          H-1
<Feb 18 15:11:16> ->CMCONV
<Feb 18 15:11:16> <-CMCONV - 0.000
<Feb 18 15:11:16> ->CMCONV
<Feb 18 15:11:16> <-CMCONV - 0.050
<Feb 18 15:11:16> ->CMCONV
<Feb 18 15:11:16> <-CMCONV - 0.000
<Feb 18 15:11:16> ->CMCONV
<Feb 18 15:11:16> <-CMCONV - 0.000
<Feb 18 15:11:16> ==0.220
<Feb 18 15:11:16> <-DCE:CMDV_CALL
<Feb 18 15:11:16><-CMCALL - 5.930          F-3
<Feb 18 15:11:19>->CMCALL
<Feb 18 15:11:19> Calling application VGD2AIX B-4
<Feb 18 15:11:19> ->readFromLinkTbl      C-4
<Feb 18 15:11:19> <-readFromLinkTbl - 0.000
<Feb 18 15:11:19> ->DCE:CMDV_CALL
<Feb 18 15:11:19> ++DCE:CMDV_CALL
<Feb 18 15:11:19> ++++DCE:Convert          G-2
<Feb 18 15:11:19> ->CMCONV
<Feb 18 15:11:19> <-CMCONV - 0.060
<Feb 18 15:11:19> ->CMCONV
<Feb 18 15:11:19> <-CMCONV - 0.000
<Feb 18 15:11:19> ->CMCONV
<Feb 18 15:11:19> <-CMCONV - 0.000
<Feb 18 15:11:19> ->CMCONV
<Feb 18 15:11:19> <-CMCONV - 0.060
<Feb 18 15:11:19> ===0.170
<Feb 18 15:11:19> ++++DCE:CreateParmBlock E-4
<Feb 18 15:11:19> ===0.000
<Feb 18 15:11:19> Using binding handle: f80bd000-89e3-11d0-a4e4-10005ab1f3f8@ncadg_ip_udp:9.67.172.228[]
<Feb 18 15:11:19> Passing 428 bytes of data
<Feb 18 15:11:19> ++++DCE:RPCCall
<Feb 18 15:11:20> ===0.660
<Feb 18 15:11:20> ==0.990
<Feb 18 15:11:20> ++DCE:unConvert          H-2
<Feb 18 15:11:20> ->CMCONV
<Feb 18 15:11:20> <-CMCONV - 0.000
<Feb 18 15:11:20> ->CMCONV
<Feb 18 15:11:20> <-CMCONV - 0.050
<Feb 18 15:11:20> ->CMCONV
<Feb 18 15:11:20> <-CMCONV - 0.060
<Feb 18 15:11:20> ->CMCONV
<Feb 18 15:11:20> <-CMCONV - 0.000
<Feb 18 15:11:20> ==0.220
<Feb 18 15:11:20> <-DCE:CMDV_CALL
<Feb 18 15:11:20><-CMCALL - 1.320          F-4

```

Figure 60 (Part 2 of 2). VisualAge Generator Windows 95 Client Trace Log for DCE-Based Client/Server Communication Calls

In Figure 60 on page 81, the following processing points should be understood:

- A** One-time initialization of the client/server communication environment.
- B** Identification of the application being called.
- C** Obtaining runtime binding information from linkage table.
- D** One-time initialization of client/server communication service driver.
- E** Formatting parameters for DCE-based client/server communication call. This is followed by a string that contains the TCP/IP address for the DCE application server workstation.
- F** Return to client application with time for server call. The time required for the first call to each target platform includes the lookup of the DCE destination and the target platform runtime initialization. (On the OS/2 target platform, the **F-1** entry includes DB2/2 database startup processing.) The DCE target location is cached on the client so lookups are not required for subsequent calls.
- G** Conversion of binary data to AIX platform format prior to call.

H

Conversion of binary data from AIX platform format after call.

Note: Server responsiveness will depend on hardware capability, software tuning, and network latency. We were not using production-quality hardware platforms. Your results will differ, because they will be based on your hardware processor and memory as well as the active network configuration. We saw different response times on different days due to changes in the network workload.

With the DCE-based client/server communication configuration we implemented, multiple call paths could be tested (see Figure 61).

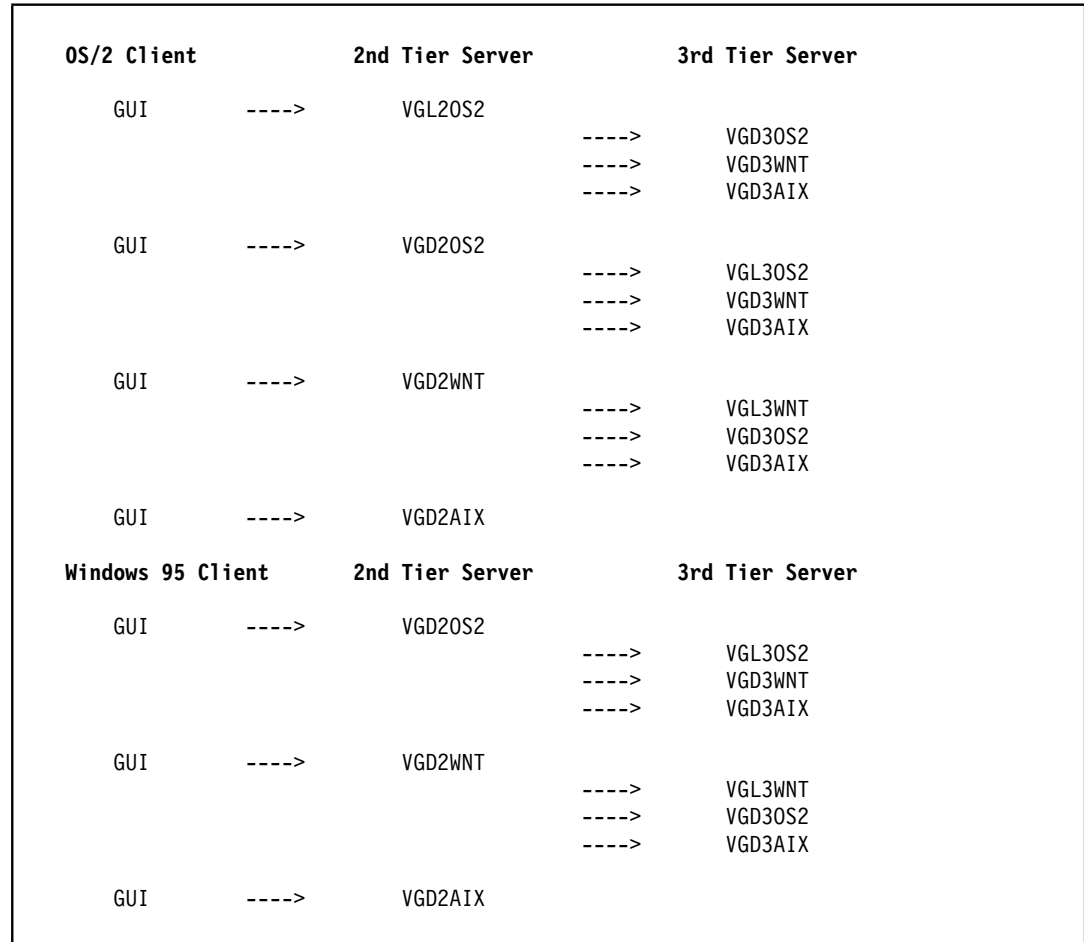


Figure 61. DCE-Based Client/Server Communication Call Path Options

We had one CSODCES DCE server program running for each target platform (LOCATION values: vgos2, vgwn2, and vgaix). Our testing showed that a DCE-based VisualAge Generator server application cannot call another DCE-based server that is located in the same LOCATION. That is, VG20S2 can not call VGD30S2. When we attempted this type of call on any target platform, the sample application hung up.

A DCE-based server can call a local server (for example: VG20S2 can call VGL30S2 and VG2WNT can call VGL3WNT).

4.2.7.4 Implementing Multiple DCE Application Server Workstations

By starting a secondary DCE application server program on an additional workstation with the same SERVERID/LOCATION pair that is configured on the primary workstation, you allow DCE to balance the calls for the same set of VisualAge Generator server applications across multiple application-server workstations. This can improve system performance, provide a form of redundancy, and support a scalable system infrastructure.

Note: During our testing we could not implement more than one CSODCES DCE server program for a specific SERVERID/LOCATION pair on a single workstation. When we started more than one CSODCES DCE server program for a SERVERID/LOCATION pair, calls were routed solely to the DCE server program that was started first.

The parameters for CSODCES are:

-c or -d This is the optional cleanup parameter. The cleanup parameter is required when more than one server is using a SERVERID/LOCATION pair as its advertising location. The -c value is the documented default.¹⁶

When -c is used, a primary DCE server program is started. During termination processing, a primary DCE server program will remove entries from the RPC mapping, DCE runtime, and DCE CDS. When these entries are removed, no servers can be called for a given SERVERID/LOCATION pair.

When -d is used, a secondary DCE server program is started. During termination processing, a secondary DCE server program removes only RPC mapping entries. This prevents calls from being routed to a particular application server workstation. Other active application server workstations for a given SERVERID/LOCATION pair can still process server calls.

configuration-file name

The configuration file specifies which VisualAge Generator server applications are available (can be called) on the application server workstation. The VisualAge Generator server applications available for one application server workstation must be available for all other application server workstations that are started for a given SERVERID/LOCATION pair.

We added a second OS/2 application server workstation to the configuration shown in Figure 50 on page 75. The startup messages for the primary DCE server program on the first application server workstation are shown in Figure 62 on page 85.

¹⁶ There is a problem with VisualAge Generator V2.2 at the Fixpak 3 level. The documented default parameter (-c) must be used to trigger primary server processing. If -c is not provided, the CSODCES program processes as if the -d parameter was provided, which makes the -d parameter the effective default. The -d parameter, when used, causes an error message (Could not open data file, -d).

```
[E:\vgcs\dcesec]csodces -c vgos2.cnf
DCEprincipal TOKEN=vgos2sj1
LOCATION TOKEN=vgos2
SERVERID TOKEN=vgcs
DCEACLOBJECT TOKEN=./Servers/VAGenerator/vgcs/vgos2
SECURE APPLICATIONS =
PUBLIC APPLICATIONS =
VGD2OS2
VGD3OS2
Server to be loaded VGD2OS2.
Server to be loaded VGD3OS2.
Registering server interface with RPC runtime...
Registering server endpoints with endpoint mapper (RPCD)...
Exporting server bindings into CDS namespace...
Server ./Servers/VAGenerator/vgcs/vgos2 listening...
```

Figure 62. Primary DCE Server Program Startup Messages. The documented default when a startup option is not specified is **-c**. The **-c** option is specified because of a bug in VisualAge Generator V2.2 at the Fixpak 3 level, where **-d** is the effective default. This problem was reported to the IBM VisualAge Generator lab.

The startup messages for the secondary DCE server program on the second application server workstation are shown in Figure 63.

```
[S:\pat\vgcs-run\dce]csodces vgos22.cnf
DCEprincipal TOKEN=vgos2sj2
LOCATION TOKEN=vgos2
SERVERID TOKEN=vgcs
DCEACLOBJECT TOKEN=./Servers/VAGenerator/vgcs/vgos2
SECURE APPLICATIONS =
PUBLIC APPLICATIONS =
VGD2OS2
VGD3OS2
Server to be loaded VGD2OS2.
Server to be loaded VGD3OS2.
Registering server interface with RPC runtime...
Registering server endpoints with endpoint mapper (RPCD)...
Exporting server bindings into CDS namespace...
Server ./Servers/VAGenerator/vgcs/vgos2 listening...
```

Figure 63. Secondary DCE Server Program Startup Messages. Because of a bug in VisualAge Generator V2.2 at the Fixpak 3 level, **-d** is the effective default. If **-d** is specified, an error message results. This problem was reported to the IBM VisualAge Generator lab.

The CDS object **vgos2**, used as the **LOCATION** value for both the primary and secondary DCE server programs, is used to locate where the requested VisualAge Generator server applications can be found. Using a DCE command we can see that both workstations are identified in the information stored in the CDS object by DCE (see Figure 64 on page 86).

```

cdscp> show object ../Servers/VAGenerator/vgcs/vgos2

      SHOW
      OBJECT   /.../vgrisc/Servers/VAGenerator/vgcs/vgos2
      AT       1997-02-21-19:14:13
RPC_ClassVersion = 0100
RPC_ObjectUUIDs = 80d3f926e489d0119b3708005a94dcc5
RPC_ObjectUUIDs = 602a1728e489d0119b3708005a94dcc5
      CDS_CTS = 1997-02-20-18:05:20.371582100/10-00-5a-b1-f3-f8
      CDS_UTS = 1997-02-22-00:13:25.767033100/10-00-5a-b1-f3-f8
      CDS_Class = RPC_Server
CDS_ClassVersion = 1.0
  CDS_Towers = :
    Tower = ncadg_ip_udp:129.33.160.215[]
  CDS_Towers = :
    Tower = ncadg_ip_udp:129.33.160.233[]

cdscp>

```

Figure 64. CDS Object Information for Two Application Server Workstations. The highlighted tower data includes the TCP/IP address for both the primary and the secondary OS/2 application server workstations.

When multiple application-server workstations are available, the choice of which one is used to satisfy a call is randomized. Figure 65 shows the client/server communication trace log for a client where the same server call is resolved by two different workstations.

```

<Feb 21 17:03:01>->CMCALL
<Feb 21 17:03:01> Calling application V6D20S2
<Feb 21 17:03:01> ->readFromLinkTbl
<Feb 21 17:03:01> <-readFromLinkTbl - 0.050
<Feb 21 17:03:01> ->DCE:CMDV_CALL
<Feb 21 17:03:01> ++DCE:CMDV_CALL
<Feb 21 17:03:01> ++++DCE:CreateParmBlock
<Feb 21 17:03:01> ====0.060
<Feb 21 17:03:01> Using binding handle: 26f9d380-89e4-11d0-9b37-08005a94dcc5@ncadg_ip_udp:129.33.160.233[]
<Feb 21 17:03:01> Passing 428 bytes of data
<Feb 21 17:03:01> ++++DCE:RPCCall
<Feb 21 17:03:06> ====4.770
<Feb 21 17:03:06> ==5.160
<Feb 21 17:03:06> <-DCE:CMDV_CALL
<Feb 21 17:03:06><-CMCALL - 5.490
<Feb 21 17:03:07>->CMCALL
<Feb 21 17:03:07> Calling application V6D20S2
<Feb 21 17:03:07> ->readFromLinkTbl
<Feb 21 17:03:07> <-readFromLinkTbl - 0.060
<Feb 21 17:03:07> ->DCE:CMDV_CALL
<Feb 21 17:03:07> ++DCE:CMDV_CALL
<Feb 21 17:03:07> ++++DCE:CreateParmBlock
<Feb 21 17:03:08> ====0.050
<Feb 21 17:03:08> Using binding handle: 26f9d380-89e4-11d0-9b37-08005a94dcc5@ncadg_ip_udp:129.33.160.215[]
<Feb 21 17:03:08> Passing 428 bytes of data
<Feb 21 17:03:08> ++++DCE:RPCCall
<Feb 21 17:03:10> ====2.630
<Feb 21 17:03:10> ==2.970
<Feb 21 17:03:10> <-DCE:CMDV_CALL
<Feb 21 17:03:10><-CMCALL - 3.240
<Feb 21 17:03:13>->CMCOMMIT
<Feb 21 17:03:13><-CMCOMMIT - 0.060
<Feb 21 17:03:13>->CMCLOSE
<Feb 21 17:03:13><-CMCLOSE - 0.000

```

Figure 65. Client/Server Communication Trace Log for Multiple Application Server Workstations

With the -c parameter, when the primary DCE server program CSODCES terminates, it will remove all DCE binding information for the active SERVERID/LOCATION pair. This means that secondary servers started to support the same SERVERID/LOCATION would still be running, but not accessible.

Secondary servers should be started with the *-d* parameter. When you stop a version of the CSODCES program that was started with the *-d* parameter, it removes only its own entry from the DCE Advertiser as part of termination.

The rules for running multiple copies of CSODCES that advertise for the same *SERVERID/LOCATION* are these:

- Start the first (primary) DCE server with the command: `csodces -c config-file`
- Start subsequent (secondary) DCE servers with the command: `csodces -d config-file` (csodces config-file if using the FixPak 3 level of VisualAge Generator)
- Stop all secondary DCE servers (those started with the *'-d'* option) first
- Stop the primary DCE server (the one started with the *'-c'* option) last.

The shutdown messages for the secondary DCE server program on the second application server workstation are shown in Figure 66.

```
[S:\pat\vgcs-run\dce]csodces vgos22.cnf
DCEprincipal TOKEN=vgos2sj2
LOCATION TOKEN=vgos2
SERVERID TOKEN=vgcs
DCEACLOBJECT TOKEN=../Servers/VAGenerator/vgcs/vgos2
SECURE APPLICATIONS =
PUBLIC APPLICATIONS =
VGD20S2
VGD30S2
Server to be loaded VGD20S2.
Server to be loaded VGD30S2.
Registering server interface with RPC runtime...
Registering server endpoints with endpoint mapper (RPCD)...
Exporting server bindings into CDS namespace...
Server ../Servers/VAGenerator/vgcs/vgos2 listening...
Unregistering interface from EPV...

[S:\pat\vgcs-run\dce]
```

Figure 66. Secondary DCE Server Program Shutdown Messages. The highlighted statements show the secondary DCE server program removing the RPC mapping entries. This prevents calls from being routed to a particular application server workstation. Other active application server workstations for a given *SERVERID/LOCATION* pair can still process server calls.

The shutdown messages for the primary DCE server program on the first application server workstation are shown in Figure 67 on page 88.

```

[E:\vgcs\dcesec]csodces -c vgos2.cnf
DCEprincipal TOKEN=vgos2sj1
LOCATION TOKEN=vgos2
SERVERID TOKEN=vgcs
DCEACLOBJECT TOKEN=../Servers/VAGenerator/vgcs/vgos2
SECURE APPLICATIONS =
PUBLIC APPLICATIONS =
VGD2OS2
VGD3OS2
Server to be loaded VGD2OS2.
Server to be loaded VGD3OS2.
Registering server interface with RPC runtime...
Registering server endpoints with endpoint mapper (RPCD)...
Exporting server bindings into CDS namespace...
Server ../Servers/VAGenerator/vgcs/vgos2 listening...
Unexporting the binding information from the namespace...
Unregistering interface from RPC runtime...
Unregistering interface from EPV...

[E:\vgcs\dcesec]

```

Figure 67. Primary DCE Server Program Shutdown Messages. The highlighted statements show the primary DCE server program removing entries from the RPC mapping, DCE run time, and DCE CDS. When these entries are removed, no servers can be called for a given SERVERID/LOCATION pair.

4.2.8 Implementing Secure DCE-Based Client/Server Communication

We are already prepared for the implementation of security for our DCE-based client/server communication environment. Earlier, as described in 4.2.2.2, “DCE Environment Configuration” on page 60, we

- Explicitly permitted the required access to the base DCE directory used by the VisualAge Generator DCE server program (../Servers/VAGenerator) to the vgservers group (see Figure 38 on page 63).
This ensured that only authorized DCE server programs could modify the directory structure and create objects in the base directory used by VisualAge Generator DCE-based client/server communication.
- Altered the default access for unauthenticated and any_other DCE users for the base DCE directory used by the VisualAge Generator DCE server program (see Figure 39 on page 64).
This prevented any users, even those not logged in to the DCE cell, from having test authority on objects created in a SERVERID directory. The test authority is required for successful REMOTECOMTYPE=dcesecure client/server communication processing.

These changes did not implement security, but they did prepare the directory structure so that security could be implemented.

With the appropriate DCE ACL definitions and DCE-based client/server communication configuration file, we can implement two forms of security:

Logon required

Access to VisualAge Generator server applications is not controlled, but the user must at least be logged on to DCE.

This can be implemented using the REMOTECOMTYPE=dce linkage table option and the right ACL definitions for the DCE CDS objects.

Authorization required

Only specifically identified and logged on DCE users can access the available VisualAge Generator server applications.

This can be implemented using the REMOTECOMTYPE=dcesecure linkage table option and the right ACL definitions for the DCE CDS objects.

4.2.8.1 Forcing Clients to be Logged on to the DCE Cell

With the existing configuration we could support DCE-based client/server communication, but none of the client workstations had to be logged on to the DCE Cell.

This anomaly occurs because the ACL definitions for the CDS directory structure permitted unauthenticated users (those not logged on to any DCE Cell) read access. All VisualAge Generator needs to support REMOTECOMTYPE=dce client/server communication processing is read access to the objects in the SERVERID directory.

We can use the DCE `acl_edit` command to check the current ACL entries for the SERVERID directory (see Figure 68).

```
vgrisc:/> acl_edit ../Servers/VAGenerator/vgcs -l

# SEC_ACL for ../Servers/VAGenerator/vgcs:
# Default cell = ../vgrisc
unauthenticated:r-----
user:cell_admin:rwdtcia
user:hosts/vgrisc/cds-server:rwdtcia
group:subsys/dce/cds-admin:rwdtcia
group:subsys/dce/cds-server:rwdtcia
group:vgsservers:rwdtci-
any_other:r-----

vgrisc:/> acl_edit ../Servers/VAGenerator/vgcs -io -l

# Initial SEC_ACL for objects created under: ../Servers/VAGenerator/vgcs:
# Default cell = ../vgrisc
unauthenticated:r-----
group:subsys/dce/cds-admin:rwdtc--
group:subsys/dce/cds-server:rwdtc--
group:vgsservers:rwdtci-
any_other:r-----

vgrisc:/> acl_edit ../Servers/VAGenerator/vgcs -ic -l

# Initial SEC_ACL for directories created under: ../Servers/VAGenerator/vgcs:
# Default cell = ../vgrisc
unauthenticated:r-----
group:subsys/dce/cds-admin:rwdtcia
group:subsys/dce/cds-server:rwdtcia
group:vgsservers:rwdtci-
any_other:r-----
```

Figure 68. Checking Existing ACL for CDS Directory with `acl_edit`. The ACLs for the SERVERID directory (vgcs), those for objects created in this directory (-io option), and those for directories created in this directory (-ic option) are shown. The authority granted to unauthenticated users is what allows clients to call servers without logging on to the DCE Cell.

The ACL entries shown in Figure 68 define the access for the directory and any objects or directories created in that directory. The LOCATION object created by the DCE server program CSODCES obtains its access authority from the initial object (-ic `acl_edit` option) entry. The LOCATION object has an ACL entry that matches the initial object entry for the directory the object is created in, at the time the object is created. Figure 69 on page 90 shows the ACL entry for the vgos2 LOCATION object created for our OS/2 application server workstation.

```

vgrisc:/> acl_edit -e ./:/Servers/VAGenerator/vgcs/vgos2 -1

# SEC_ACL for ./:/Servers/VAGenerator/vgcs/vgos2:
# Default cell = /.../vgrisc
unauthenticated:r----
user:vgos2sj2:rwdtc
group:subsys/dce/cds-admin:rwdtc
group:subsys/dce/cds-server:rwdtc
group:vgsservers:rwdtc
any_other:r----

```

Figure 69. Initial ACL Entry for LOCATION Object. The authority granted to unauthenticated users allows clients to call servers without logging on to the DCE Cell.

To force client platforms to log on to the DCE Cell, we must restrict access to the SERVERID directory to logged-on users. The following steps are required to restrict VisualAge Generator application server access to logged-on client workstations:

1. Remove unauthenticated user read access to the SERVERID directory.
Figure 70 contains the `acl_edit` commands used to remove the unauthenticated user ACL entry.

```

acl_edit ./:/Servers/VAGenerator/vgcs -d unauthenticated:r-----
acl_edit ./:/Servers/VAGenerator/vgcs -io -d unauthenticated:r-----
acl_edit ./:/Servers/VAGenerator/vgcs -ic -d unauthenticated:r-----

```

Figure 70. `acl_edit` Commands to Remove Unauthenticated User Access

This does not change the ACL for existing objects. ACLs for objects cannot be directly changed. You must change the initial object ACL for the directory where the object is to be created and then create (or recreate) the object. This leads to our next step.

2. Stop any active DCE server programs and delete the existing LOCATION and VisualAge Generator application server objects from the SERVERID directory.
Figure 71 contains the `cdscp` commands used to list and then delete the existing objects in the SERVERID directory.

```

vgrisc:/> cdscp
cdscp> list obj ./:/Servers/VAGenerator/vgcs/*
          LIST
OBJECT    /.../vgrisc/Servers/VAGenerator/vgcs
AT        1997-02-26-14:50:13

VGD20S2
VGD30S2
vgos2
cdscp>
cdscp> del obj ./:/Servers/VAGenerator/vgcs/vgos2
cdscp> del obj ./:/Servers/VAGenerator/vgcs/VGD20S2
cdscp> del obj ./:/Servers/VAGenerator/vgcs/VGD30S2
cdscp>
cdscp> list obj ./:/Servers/VAGenerator/vgcs/*
          LIST
OBJECT    /.../vgrisc/Servers/VAGenerator/vgcs
AT        1997-02-26-17:09:19

cdscp>

```

Figure 71. `cdscp` Commands to List and Delete SERVERID Directory Objects

This clears the old objects and their associated ACLs. We can now recreate the objects (we let the DCE server program do this for us).

- Restart the DCE server program (recreates LOCATION and VisualAge Generator application server objects in the SERVERID directory).

The new objects created have ACLs based on the initial object entry for the SERVERID directory `././Servers/VAGenerator/vgcs`. The LOCATION object now has the ACL entries required to support read access by clients that have logged on to the DCE Cell (see Figure 72).

```
vgrisc:/> acl_edit -e ././Servers/VAGenerator/vgcs/vgos2 -l

# SEC_ACL for ././Servers/VAGenerator/vgcs/vgos2:
# Default cell = ../../vgrisc
user:vgos2sj2:rwtdtc
group:subsys/dce/cds-admin:rwtdtc
group:subsys/dce/cds-server:rwtdtc
group:vgsservers:rwtdtc
any_other:r----
vgrisc:/>
```

Figure 72. ACL Entry List for Location Object to Force DCE Login

- Test VisualAge Generator application server access without logging on to the DCE Cell (and after logging on to the DCE Cell)

You can use the `acl_edit` command to predict the success of the server call. Since you need read access to the object, you can ask if you can in fact read the object before logging on to the DCE Cell (see Figure 73).

```
[S:\pat\vgcs-run\os2]acl_edit ././Servers -l
Warning - you currently have no tickets
Warning - binding to ACL's server is unauthenticated
Warning - binding to registry is unauthenticated

# SEC_ACL for ././Servers:
# Default cell = ../../vgrisc
unauthenticated:r--t---
user:cell_admin:rwtdcia
user:hosts/vgrisc/cds-server:rwtdcia
group:subsys/dce/cds-admin:rwtdcia
group:subsys/dce/cds-server:rwtdcia
any_other:r--t---

[S:\pat\vgcs-run\os2]acl_edit ././Servers/VAGenerator -l
ERROR: acl object not found (dce / sec)
Unable to bind to object ././Servers/VAGenerator

[S:\pat\vgcs-run\os2]
```

Figure 73. Testing Directory Access for Unauthenticated DCE Users with `acl_edit` Commands

When a call to a VisualAge Generator server application is made before logging on to the DCE cell, the resulting error message can be seen in the client/server communication log (see Figure 74 on page 92).

```

<Feb 26 14:43:14><-CMINIT - 0.110
<Feb 26 14:43:14>->CMCALL
<Feb 26 14:43:14> Calling application VGD20S2
<Feb 26 14:43:14> ->readFromLinkTbl
<Feb 26 14:43:15> <-readFromLinkTbl - 0.380
<Feb 26 14:43:15> ->loadAndInitDriver
<Feb 26 14:43:16> <-loadAndInitDriver - 1.210
<Feb 26 14:43:16> ->CMDV_INIT
<Feb 26 14:43:16> <-CMDV_INIT - 0.060
<Feb 26 14:43:16> ->DCE:CMDV_CALL
<Feb 26 14:43:16> ++DCE:CMDV_CALL
<Feb 26 14:43:16> ++++DCE:CreateParmBlock
<Feb 26 14:43:16> ===0.050
<Feb 26 14:43:17> ERROR REASON CODE: 7800, Message: CS07800E An error was encountered while performing DCE API call
rpc_ns_entry_object_inq_next. The DCE error string is entry not found (dce / rpc).
<Feb 26 14:43:17> <-DCE:CMDV_CALL - 1.040
<Feb 26 14:43:19> ->CMCOMMIT
<Feb 26 14:43:19> <-CMCOMMIT - 0.050
<Feb 26 14:43:19> ->CMCLOSE
<Feb 26 14:43:19> <-CMCLOSE - 0.050

```

Figure 74. Trace Log for Unauthenticated VisualAge Generator Server Application Call

Once a DCE cell logon has been performed, both the `acl_edit` commands and the client call to the VisualAge Generator server application are successful (see Figure 75 and Figure 76 on page 93).

```

[S:\pat\vgcs-run\os2]acl_edit ./Servers/VAGenerator -l

# SEC_ACL for ./Servers/VAGenerator:
# Default cell = ../vgrisc
user:cell_admin:rwdtcia
user:hosts/vgrisc/cds-server:rwdtcia
group:subsys/dce/cds-admin:rwdtcia
group:subsys/dce/cds-server:rwdtcia
group:vgservers:rwdtci-
any_other:r-----

[S:\pat\vgcs-run\os2]acl_edit -e ./Servers/VAGenerator/vgcs/vgos2 -l

# SEC_ACL for ./Servers/VAGenerator/vgcs/vgos2:
# Default cell = ../vgrisc
user:vgos2sj2:rwdtc
group:subsys/dce/cds-admin:rwdtc
group:subsys/dce/cds-server:rwdtc
group:vgservers:rwdtc
any_other:r----

[S:\pat\vgcs-run\os2]

```

Figure 75. Testing Directory Access for Authenticated DCE Users with `acl_edit` Commands

```

<Feb 26 16:00:01>->CMCALL
<Feb 26 16:00:01> Calling application VGD20S2
<Feb 26 16:00:01> ->readFromLinkTbl
<Feb 26 16:00:01> <-readFromLinkTbl - 0.050
<Feb 26 16:00:01> ->DCE:CMDV_CALL
<Feb 26 16:00:01> ++DCE:CMDV_CALL
<Feb 26 16:00:01> ++++DCE:CreateParmBlock
<Feb 26 16:00:01> ===0.060
<Feb 26 16:00:01> Using binding handle: 0ab5fbc0-9026-11d0-9956-08005a94dcc5@ncadg_ip_udp:129.33.160.215[]
<Feb 26 16:00:01> Passing 428 bytes of data
<Feb 26 16:00:01> ++++DCE:RPCCall
<Feb 26 16:00:04> ===2.640
<Feb 26 16:00:04> ==2.960
<Feb 26 16:00:04> <-DCE:CMDV_CALL
<Feb 26 16:00:04><-CMCALL - 3.240

```

Figure 76. Trace Log for Authenticated VisualAge Generator Server Application Call

By using native DCE CDS authorizations, as defined in the directory and object ACLs, we can implement a form of security. All client workstations must have an active logon to the DCE Cell to call a VisualAge Generator server application.

We did not restrict who had read access to the SERVERID directory and LOCATION object; we allowed *any_other* users access. We could have used a DCE group to control who had read access and removed the ACL entry for *any_other* users. This would add another level of security to the system.

By using only DCE functions, we have not added any overhead to the call of a VisualAge Generator server application. This differs from the next technique, which asks the DCE server program to validate that the DCE user requesting the VisualAge Generator server application has a specific level of authority before the call is processed.

4.2.8.2 Restricting Servers to Authorized Users with DCESECURE

The VisualAge Generator DCE-based client/server communication option REMOTECOMTYPE=dcesecure adds additional control and resource checking to the application system. This includes cyclic redundancy checking on the data sent over the wire and authorization checking for each server call.

Note: It is possible to specify REMOTECOMTYPE=dcesecure in the linkage table for *public* applications. If this is done, then the application data passed to the server will receive cyclic redundancy checking for possible corruption during transmission, but authorization checking is not performed.

The DCE server program uses the ACL identified on the DCEACLobject statement in the DCE configuration file for authorization checking:

```
DCEACLobject=././Servers/VAGenerator/vgcs/vgos2
```

(See 4.2.7.1, “DCE Server Program Configuration” on page 75 for a DCE configuration file description.)

The DCE server program checks whether the DCE client making the server call has the test (t) authority for the identified ACL object to determine if the DCE client can have access to a secure VisualAge Generator server application.

The DCE server program CSODCES will only allow access to a secure VisualAge Generator server application when both of the following conditions are true:

- The user’s DCE principal ID, or a DCE group that contains the principal ID as a member, must be included as an entry in the ACL for the security object.

- The authorization list for the ACL entry for the principal or group must include the test authority. (They must also be have read (r) access to the security object if it is the same as the LOCATION object—which is true for our examples.)

If we had not revised the ACL entries for our base DCE CDS directory (see 4.2.2.2, “DCE Environment Configuration” on page 60), then test authority, and therefore secure server access, would be allowed to these DCE users:

unauthenticated	Users not logged on to DCE
any_other	Any logged-on DCE user

In other words, there would not be any security.

Our current ACL entries for the LOCATION object (see Figure 75 on page 92) do not permit the test authority to the unauthenticated or any other DCE user categories. Specific authorities are not permitted to any of the end user principal IDs (vguserx or vgadx) or groups (vgusers, vgusradm)¹⁷ we defined for our DCE-based client/server communication environment (see Figure 34 on page 61). We are now ready to implement active authorization checking for a configuration that permits only members of the vgusradm group to call VisualAge Generator server applications.

To implement DCE server program authorization checking, we must provide the test authority on the security object (defined to be the same as our LOCATION object) to our selected set of users (the vgusradm group from Figure 34 on page 61), configure the DCE-based client/server communication environment to use the REMOTECOMTYPE=dcesecure linkage table option, and log on to the DCE Cell with an authorized principal ID.

The following steps were performed to implement active authorization checking of client workstation requests for VisualAge Generator server applications on the OS/2 application server workstation:

1. Add an entry for the group vgusradm to the initial object ACL for the SERVERID directory

Figure 77 contains the acl_edit commands used to add an initial object ACL entry to, and then list the entries for, the SERVERID directory for the vgusradm group.

```
acl_edit ../Servers/VAGenerator/vgcs -io -m group:vgusradm:rt
vgrisc:/> acl_edit ../Servers/VAGenerator/vgcs -io -l

# Initial SEC_ACL for objects created under: ../Servers/VAGenerator/vgcs:
# Default cell = ../vgrisc
group:subsys/dce/cds-admin:rwtdc--
group:subsys/dce/cds-server:rwtdc--
group:vgservers:rwtdci-
group:vgusradm:r--t---
any_other:r-----
vgrisc:/>
```

Figure 77. acl_edit Commands to Permit Test Access for vgusradm Group

To change the ACL for the LOCATION object we must delete and then create the object. This leads to our next step.

2. Stop any active DCE server programs and delete the existing LOCATION object from the SERVERID directory

¹⁷ We do permit access to the vgservers group. This group represents DCE server program tasks. These are permitted access to the directory structure so that location and VisualAge Generator server application objects can be created. See Figure 38 on page 63 for details.

Figure 78 on page 95 contains the cscdp commands used to delete the LOCATION object and then list the remaining objects in the SERVERID directory.

```
vgrisc:/> cdscp del obj ./:/Servers/VAGenerator/vgcs/vgos2
vgrisc:/> cdscp list obj ./:/Servers/VAGenerator/vgcs/*
LIST
OBJECT  /.../vgrisc/Servers/VAGenerator/vgcs
AT      1997-02-28-17:49:33

VGD20S2
VGD30S2
vgrisc:/>
```

Figure 78. cdscp Commands to Delete LOCATION Object and then List Remaining SERVERID Directory Objects

We deleted the the LOCATION object so that it could be recreated with a new set of ACL entries based on the initial object ACL defined for the SERVERID directory.

3. Change the linkage table entries for DCE-based VisualAge Generator server applications so that calls use the REMOTECOMTYPE=dcsecure communications option.
The linkage table entries for calls to the VisualAge Generator server applications on the OS/2 application server workstation are shown in Figure 79.

```
:CALLLINK APPLNAME=VGD20S2 LIBRARY=VGD20S2 REMOTECOMTYPE=dcsecure
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgos2 REMOTEbind=runtime.

:CALLLINK APPLNAME=VGD20S2 LIBRARY=VGD30S2 REMOTECOMTYPE=dcsecure
          PARMFORM=osl link LINKTYPE=remote LUWCONTROL=server
          SERVERID=vgcs LOCATION=vgos2 REMOTEbind=runtime.
```

Figure 79. Linkage Table for OS/2 Sample Application Servers with Secure DCE-Based Client/Server Communication Support

These linkage table entries must be used by both the client application and the DCE server program.

4. Define a configuration file for secure communication and restart the DCE server program (recreates LOCATION and VisualAge Generator application server objects in the SERVERID directory).
The startup messages for a DCE server program configured to use secure communications are shown in Figure 80 on page 96.

```
[S:\pat\vgcs-run\dce]csodces -c vgos2sec.cnf
DCEprincipal TOKEN=vgos2sj2
LOCATION TOKEN=vgos2
SERVERID TOKEN=vgcs
DCEACLOBJECT TOKEN=./Servers/VAGenerator/vgcs/vgos2
SECURE APPLICATIONS =
VGD20S2
VGD30S2
PUBLIC APPLICATIONS =
Server to be loaded VGD20S2.
Server to be loaded VGD30S2.
Registering server interface with RPC runtime...
Registering server endpoints with endpoint mapper (RPCD)...
Exporting server bindings into CDS namespace...
Server ./Servers/VAGenerator/vgcs/vgos2 listening...
```

Figure 80. Secure DCE Server Program Startup Messages

The new objects created have ACLs based on the initial object entry for the SERVERID directory ./Servers/VAGenerator/vgcs. The LOCATION object now has an ACL entry that gives the vgusradm group read and test access (see Figure 81).

```
vgisc:/> acl_edit -e ./Servers/VAGenerator/vgcs/vgos2 -l

# SEC_ACL for ./Servers/VAGenerator/vgcs/vgos2:
# Default cell = ../vgisc
user:vgos2sj2:rwdtc
group:subsys/dce/cds-admin:rwdtc
group:subsys/dce/cds-server:rwdtc
group:vgsservers:rwdtc
group:vgusradm:r--t-
any_other:r----
vgisc:/>
```

Figure 81. ACL Entry List for Location Object with vgusradm Group

5. Test VisualAge Generator application server access with a general user logon and then with an administrative user logon.

When a call to a VisualAge Generator server application is made with a general user DCE logon, the DCE server program checks whether or not the user has test authority on the security object. Because the general users are not in the vgusradm group, their calls fail (see Figure 82 on page 97).

```

<Feb 28 16:23:05>->CMCALL
<Feb 28 16:23:05> Calling application VGD20S2
<Feb 28 16:23:05> ->readFromLinkTbl
<Feb 28 16:23:05> <-readFromLinkTbl - 0.000
<Feb 28 16:23:05> ->DCE:CMDV_CALL
<Feb 28 16:23:05> ++DCE:CMDV_CALL
<Feb 28 16:23:05> ++++DCE:CreateParmBlock
<Feb 28 16:23:05> ====0.050
<Feb 28 16:23:05> Using binding handle: 0ab5fbc0-9026-11d0-9956-08005a94dcc5@ncadg_ip_udp:129.33.160.215[1066]
<Feb 28 16:23:05> Passing 428 bytes of data
<Feb 28 16:23:05> ++++DCE:RPCCall
<Feb 28 16:23:06> ====0.870
<Feb 28 16:23:06> ERROR REASON CODE: 7814, Message: CS07814E The client is not authorized to
access the requested application.
==1.320
<Feb 28 16:23:06> <-DCE:CMDV_CALL
<Feb 28 16:23:06><-CMCALL - 1.650
<Feb 28 16:23:08>->CMCOMMIT
<Feb 28 16:23:08><-CMCOMMIT - 0.060
<Feb 28 16:23:09>->CMCLOSE
<Feb 28 16:23:09><-CMCLOSE - 0.000

```

Figure 82. Trace Log for Rejected Authenticated VisualAge Generator Server Application Call

Once an administrative user logon has been performed, the client call to the VisualAge Generator server application is successful (see Figure 83).

```

<Feb 28 16:24:09>->CMCALL
<Feb 28 16:24:09> Calling application VGD20S2
<Feb 28 16:24:09> ->readFromLinkTbl
<Feb 28 16:24:09> <-readFromLinkTbl - 0.050
<Feb 28 16:24:09> ->DCE:CMDV_CALL
<Feb 28 16:24:09> ++DCE:CMDV_CALL
<Feb 28 16:24:09> ++++DCE:CreateParmBlock
<Feb 28 16:24:09> ====0.060
<Feb 28 16:24:09> Using binding handle: 0ab5fbc0-9026-11d0-9956-08005a94dcc5@ncadg_ip_udp:129.33.160.215[1066]
<Feb 28 16:24:09> Passing 428 bytes of data
<Feb 28 16:24:09> ++++DCE:RPCCall
<Feb 28 16:24:13> ====3.840
<Feb 28 16:24:13> ==4.180
<Feb 28 16:24:13> <-DCE:CMDV_CALL
<Feb 28 16:24:13><-CMCALL - 4.510

```

Figure 83. Trace Log for Accepted Authenticated VisualAge Generator Server Application Call

You cannot control execution authority for a specific VisualAge Generator server application. You can only control access to the set of SECURE applications accessed through a DCE server program.

This means that when you are permitted test access to the security object for a DCE server program (for example DCEACLObject=/.:/Servers/VAGenerator/vgcs/vgos2), you can call all VisualAge Generator server applications defined as secure applications in the active CSODCES configuration file (as well as all the public applications). If you need to restrict access to some VisualAge Generator server applications, configure them as part of a separate LOCATION and DCE server program.

4.2.9 Common DCE-Based Client/Server Communication Configuration Error

During our environment setup and testing of different configurations we ran into many errors. These are listed below with the steps we took to resolve them. Sometimes the same error message had multiple possible causes.

4.2.9.1 SQL Errors

We only ran into one SQL problem when working with DCE-based client/server communication. Other generic DB2 problems are discussed in Table 7 on page 136.

Error message	SQL0551N: <authorization-ID> does not have the privilege to perform operation <operation> on object <name>.
Situation and Resolution	The user ID and password used did not have the required authority. Grant authority to access the object or execute the package.

4.2.9.2 DCE Setup and Configuration Errors

The setup and configuration errors we encountered are reviewed in Table 5.

Table 5 (Page 1 of 2). Common Problems Encountered During DCE Setup and Configuration
Error Message / Situation and Resolution
Error: OS/2 DCE Client startup failure. Messages included text like this: <pre>1997-01-21-16:16:43.560-08:00I----- dced ERROR dhd secval D:\U\BUILD\BUILD\SRC\ADMIN\DCED\SERVER\SV_CLIENTD.C 282 0x00377af8 msgID=0x113DB2BE Call to a sec_login_xxx function failed, status=0x17122080 1997-01-21-16:16:54.000-08:00I----- dced ERROR dhd secval D:\U\BUILD\BUILD\SRC\ADMIN\DCED\SERVER\SV_CLIENTD.C 282 0x00377af8 msgID=0x113DB2BE Call to a sec_login_xxx function failed, status=0x17122080</pre>
Situation and Resolution We never could determine exactly why DCE startup failures occurred. They were random. To resolve we unconfigured/reconfigured the OS/2 DCE Client software. When we added support for a time synchronization, these errors seemed to occur less often.
Error: CSODCES DCE server program startup failure with these startup processing messages: <pre>·E:\vgcs\dce\csodces dce.cnf DCEprincipal TOKEN=vgos2sj LOCATION TOKEN=vgos2srv SERVERID TOKEN=vgcs DCEACLOBJECT TOKEN=./Servers/VAGenerator/vgcs/vgos2srv SECURE APPLICATIONS = PUBLIC APPLICATIONS = VGD20S2 VGD30S2 Server to be loaded VGD20S2.</pre>
Situation and Resolution All the VisualAge Generator server applications must be available for loading by the CSODCES DCE server program. We had to move our generated VisualAge Generator server applications to a directory referenced by the LIBPATH environment variable.

Table 5 (Page 2 of 2). Common Problems Encountered During DCE Setup and Configuration**Error Message / Situation and Resolution****Error:**

CSODCES DCE server program startup failure with these startup processing messages:

```
•E:\vgcs\dcesec\csodces -c dce.cnf
DCEprincipal TOKEN=vgos2sj1
LOCATION TOKEN=vgos2
SERVERID TOKEN=vgcs
DCEACLOBJECT TOKEN=./Servers/VAGenerator/vgcs/vgos2
SECURE APPLICATIONS =
PUBLIC APPLICATIONS =
VGD20S2
VGD30S2
CSODCES.EXE: error in cso2dce_s.c.494':
    Requested key is unavailable (dce / sec).
```

Situation and Resolution

The DCE user ID defined in the CSODCES configuration file has not been defined in the keytab file. See Figure 54 on page 77 refid=acl. for details.

Error:

CSODCES DCE server program startup failure with these startup processing messages:

```
•E:\vgcs\dce\csodces dce.cnf
DCEprincipal TOKEN=vgos2sj
LOCATION TOKEN=vgos2srv
SERVERID TOKEN=vgcs
DCEACLOBJECT TOKEN=./Servers/VAGenerator/vgcs/vgos2srv
SECURE APPLICATIONS =
PUBLIC APPLICATIONS =
VGD20S2
VGD30S2
Server to be loaded VGD20S2.
CSODCES.EXE: error in cso2dce_s.c.590':
    No permission for name service operation (dce / rpc).
```

Situation and Resolution

The DCE user ID defined in the keytab file did not have the appropriate authority to create or modify objects in the target DCE CDS directory. We had to implement the required authority for the ID defined in the keytab file. See Figure 37 on page 62 and Figure 38 on page 63 in topic 4.2.2.2, “DCE Environment Configuration” on page 60 for details.

4.2.9.3 DCE Runtime Errors

The runtime errors we encountered are reviewed in Table 6.

Table 6 (Page 1 of 2). Common Problems Encountered During DCE Runtime	
Error Message / Situation and Resolution	
Error:	
Failed call from client to server with this text in the client/server communication CSOTRACE.OUT log:	
<pre><Jan 9 09:31:39> ->CMCALL <Jan 9 09:31:39> Calling application VGD20S2 <Jan 9 09:31:39> ->readFromLinkTbl <Jan 9 09:31:39> <-readFromLinkTbl - 0.027786 s <Jan 9 09:31:39> ->DCE:CMDV_CALL <Jan 9 09:31:39> ++++++++DCE:CMDV_CALL <Jan 9 09:31:39> ++++++++DCE:CreateParmBlock <Jan 9 09:31:39> =====0.028657 s <Jan 9 09:31:39> ERROR REASON CODE: 7800, Message: CS07800E An error was encountered while performing DCE API call rpc_ns_entry_object_inq_next. The DCE error string is Authentication ticket expired (dce / rpc). <Jan 9 09:31:39> <-DCE:CMDV_CALL - 0.166161 s</pre>	
Situation and Resolution	
We had left our client and server platforms up overnight. The authorization required to access the server logic had expired. To get things going again, we had to stop and restart the GUI runtime environment. Similar messages can occur on the CSODCES server workstation. The dynamic DCE login performed by the CSODCES server using the KEYTAB file ID expires. The CSODCES server must be stopped and started to get things going again. A requirement has been submitted to have the CSODCES program dynamically update the ID authorization so that the program can run continuously for days at a time. It is possible that advanced DCE skills could identify a method of establishing unlimited (no timeout) access for the KEYTAB defined ID in the DCE Cell. We did not investigate this option.	

Table 6 (Page 2 of 2). Common Problems Encountered During DCE Runtime**Error Message / Situation and Resolution****Error:**

Failed call from client to server with this text in the client/server communication CSOTRACE.OUT log:

```
<Feb 12 13:49:00>->CMINIT
<Feb 12 13:49:01><-CMINIT - 0.058374 s
<Feb 12 13:49:01>->CMCALL
<Feb 12 13:49:01> Calling application VGD2AIX
<Feb 12 13:49:01> ->readFromLinkTbl
<Feb 12 13:49:01> <-readFromLinkTbl - 0.215518 s
<Feb 12 13:49:01> ->loadAndInitDriver
<Feb 12 13:49:02> <-loadAndInitDriver - 0.946563 s
<Feb 12 13:49:02> ->CMDV_INIT
<Feb 12 13:49:02> <-CMDV_INIT - 0.025085 s
<Feb 12 13:49:02> ->DCE:CMDV_CALL
<Feb 12 13:49:02> ++DCE:CMDV_CALL
<Feb 12 13:49:02> ++++DCE:CreateParmBlock
<Feb 12 13:49:02> ====0.062557 s
<Feb 12 13:49:03> ERROR REASON CODE: 7801, Message: CS07801E
    An error was encountered while performing DCE API call rpc_ns_binding_import_next.
    The DCE error string is No more bindings (dce / rpc).
```

Situation and Resolution

We ran into this several times. Many times all we had to do was wait, or stop/start all client and CSODCES server program processing, or stop/start DCE client software on both client and server workstations. This can indicate a problem with the accuracy of the data in the DCE CDS cache on the client workstations.

This problem can also occur if you mix VisualAge Generator V2.2 FixPak 2 and FixPak 3 platforms in a DCE-based client/server communication configuration.

One possible method of resolution requires that you delete all objects in the SERVERID directory. This includes the LOCATION and VisualAge Generator server application objects. Once deleted stop and restart DCE on the client and server workstations and then restart the CSODCES DCE server program.

Chapter 5. Database-Enabled Client/Server

In the classic view of client/server, several approaches are defined:

Distributed presentation

The user interface is available in a remote session. CICS terminal or X-Window sessions are two examples of distributed presentation.

Remote presentation

The user interface exists on the client platform while application processing runs on a remote platform. This is a thin client approach because much of the application runs off the client workstation.

Distributed function

The user interface and some application processing occurs on the client platform and the remaining application processing runs on a remote platform. This is a heavier client with shared logic on the server.

Remote data

The user interface and all application processing occur on the client platform but the database accessed is on a remote platform. This is a fat client approach because most of the application runs on the client workstation.

Distributed data

The user interface and all application processing occurs on the client platform. Multiple databases are accessed. Databases are either a mix of local and remote or all remote. This approach is often seen as including support for updates to multiple databases in a single logical unit of work.

Note: These are simplified definitions. What makes it all more complex is that you can mix and match portions of each approach in any given system. You may call a server (remote presentation or distributed function) and the server may implement a form of remote or distributed database access.

The implementation of the remote and distributed database approach to client/server systems is supported by database technology. The application logic can be in one place, on one system, and the data involved can be accessed remotely or distributed throughout the enterprise.

While VisualAge Generator provides support for all of the client/server approaches defined above, we focus this chapter on the use of database technology. We review supported approaches to remote and distributed database access and discuss the implementation of VisualAge Generator applications that access relational databases in local, remote, and distributed environments.

5.1 Database Configuration Options

VisualAge Generator provides excellent support for the DB2 product family and, through DataJoiner, offers access to non-DB2 databases.

Several approaches to database configuration, all supported by VisualAge Generator, are reviewed in this section.

5.1.1 DB2 Common Server Connectivity

DB2 is available on multiple platforms. The current version of DB2, as available on workstation platforms, is referred to as the *DB2 Common Server*.

DB2 Common Server technology provides both stand-alone and networked relational database support. Connecting different installations of DB2 Common Server technology is very easy to do even when they are running on different workstation hardware and software platforms.

Figure 84 shows the DB2 client platforms that can be connected to other DB2 Common Server installations.

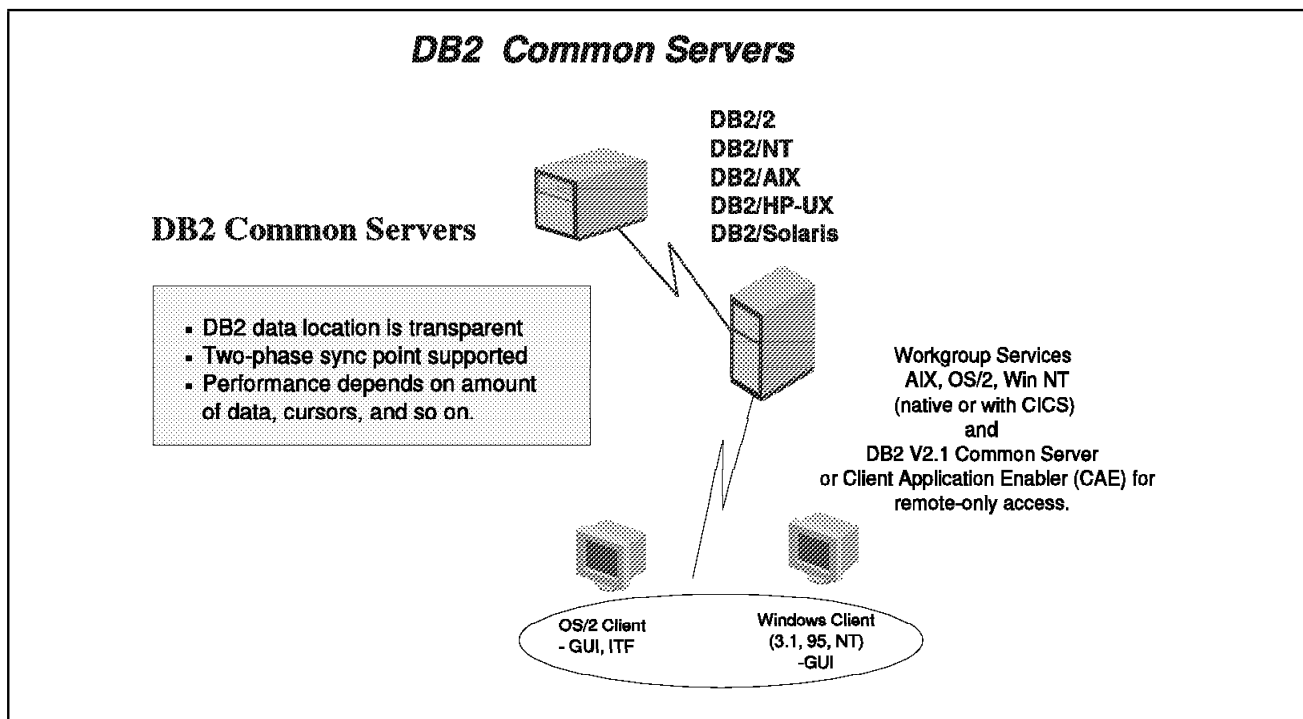


Figure 84. DB2 Common Server Configurations

5.1.2 Distributed Relational Database Architecture Connectivity

DRDA is a synchronous connection between a client running an application and a DB2 server. The client might be a user workstation or a workgroup server.

There are two performance issues you may want to consider:

- It is necessary to limit the amount of data that is passed to and from the application and the remote DB2 server.

- Heavy cursor usage is undesirable because it is more expensive over a remote connection.

Advantages of DRDA are these:

- Central database and local processing power
- If the application system has to work with a central database and with a local database, DRDA facilitates that.
- For testing during development using ITF, DRDA makes it unnecessary to download the test data.

Figure 85 shows the DB2 client platforms that can be connected to DB2 host platforms using DRDA support.

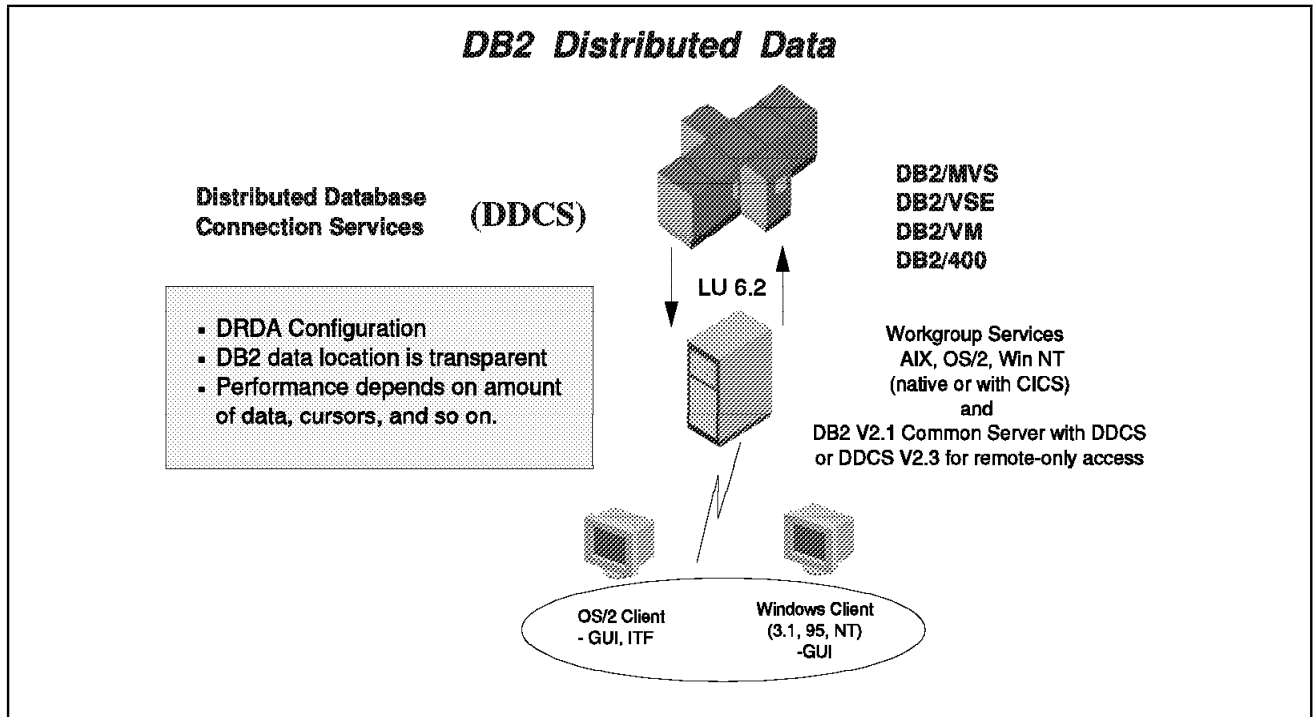


Figure 85. DB2 DRDA Configurations

Technically, you can use DRDA connections between DB2 Common Server workstations, but this is not recommended in most scenarios because of the additional software and configuration tasks required.

5.1.3 DataJoiner

Data Joiner is a an enhanced DB2/6000 that is able to catalog non-IBM relational databases. Before implementing an application system using DataJoiner, careful examinations of compatibility and performance issues are strongly recommended.

Figure 86 on page 106 shows an overview of the connectivity options supported by DataJoiner.

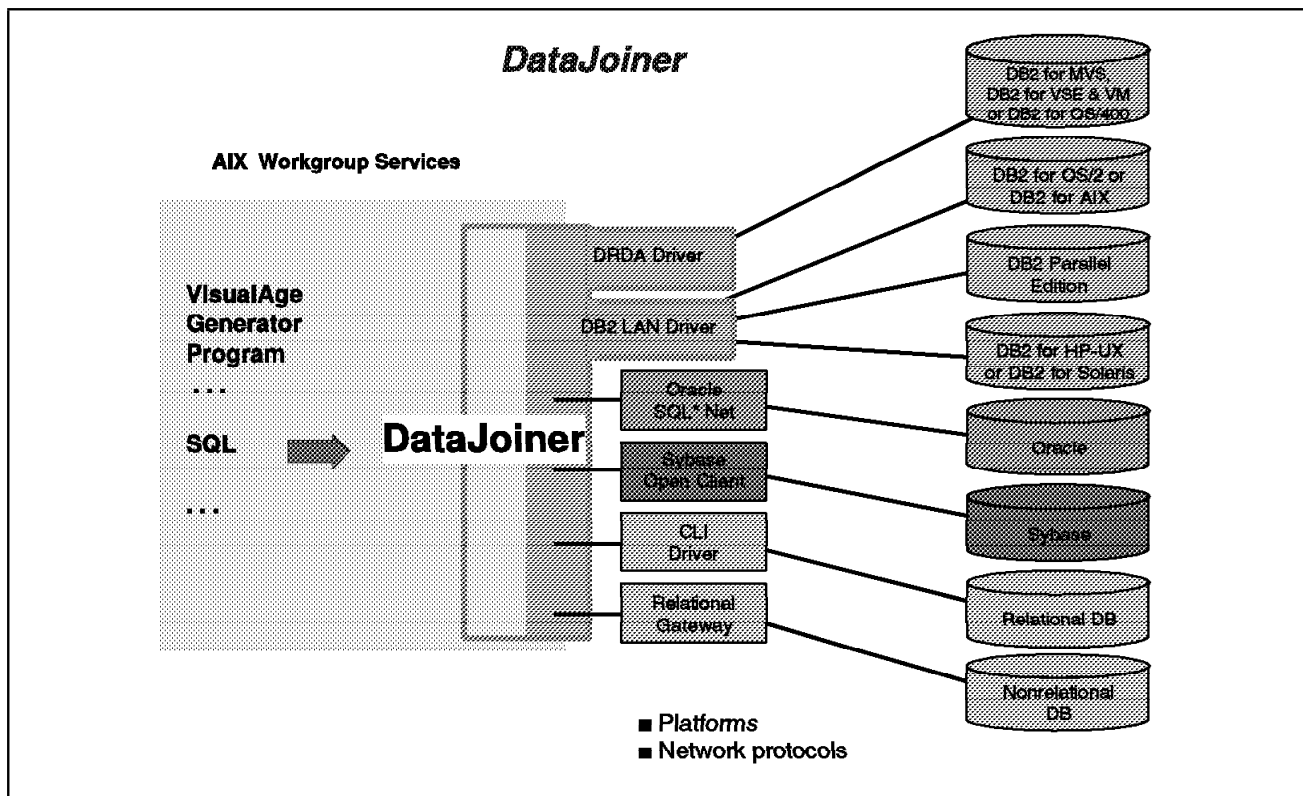


Figure 86. DataJoiner Support for Access to Non-IBM Databases

For more information, consult the DataJoiner product documentation or *DataJoiner Guide*, SG24-2566.

5.2 Accessing Relational Databases with VisualAge Generator

VisualAge Generator supports access to any local or remote DB2 database and, through the use of DataJoiner, to non-IBM database targets. This was reviewed in 5.1, "Database Configuration Options" on page 104.

This section describes the basic requirements for a working database configuration for VisualAge Generator development and runtime environments and then discuss the new support provided by VisualAge Generator Version 2.2 for distributed unit of work (DUOW) support.

5.2.1 Configuring Database Support

VisualAge Generator provides very good support for relational database access during development, generation, preparation, and execution.

Configuring VisualAge Generator for database access is very well documented in the *Running VisualAge Generator Applications on OS/2, AIX, and Windows* manual. In this section, therefore, review only the basic requirements for a functioning database connection.

5.2.1.1 Using Remote Databases

VisualAge Generator supports the use of local or remote databases.

Connecting to a local database, one that exists on the same workstation as the VisualAge Generator Developer or generated application, is straightforward. All you need to do is identify the database name.

Connecting to a remote database, one that exists on another workstation or a host platform, takes a little more work—and then you still need to identify the database name.

Remote databases can be connected using either DB2 Common Server-based or DRDA-based communications support. (We do not discuss the use of VisualAge Generator and DataJoiner in this document.)

For workstation to workstation connections, you could set up a DRDA-based connection but, in most environments, a DB2 Common Server-based connection is recommended because of the easier configuration process.

When you need to connect to a host database, you need the appropriate DDCS software and communications configuration to support a DRDA-based connection.

To enable remote database connections, you need to:

- Provide support for the selected communication protocol (TCP/IP, NetBIOS, APPC, IPX).
- Catalog the remote node.
- Catalog the remote database.
- For DRDA connections, you also need to catalog the DCS database.

Remote database configuration steps are documented in the appropriate DB2 and DDCS product manuals. For additional guidance, you might want to review *Distributed Relational Database Cross Platform Connectivity and Applications*.

5.2.1.2 Connecting to the DB2 Database

Before you configure VisualAge Generator support for relational database access, you must ensure you can access the database without VisualAge Generator.

Access to any DB2 relational base typically works if these statements are functional for your target database and a table in your database environment:

```
DB2 CONNECT TO DATABASE name
DB2 SELECT * FROM creatorid.tablename
```

The success of the connect statement depends on the availability of a valid database logon or authorization ID.

5.2.1.3 Identifying the Database Authorization ID

In an OS/2 environment, DB2 will look for an active local or LAN logon and use that user ID value for authorization processing.

In an AIX environment, the active logon ID is used for authorization checking.

For remote databases, the active local or LAN logon, if available, will be used for authorization processing unless you have logged on directly to the remote database node with a node logon:

```
LOGON userid /P=password /N=node_name
```

The `node_name` is the name of the target workstation or host where the remote database has been cataloged.

In the OS/2 environment, if a user identifier is not provided or assumed, a logon prompt is shown to request a valid local logon.

VisualAge Generator applications will use the local, node, or LAN logon for connecting to the target DB2 environment. EZECONCT statement processing can use the active DB2 authorization ID or include user ID and password values as part of the explicit connection request. This is discussed further in 5.2.2.3, "Connecting to Multiple DB2 Databases" on page 114.

5.2.1.4 Identifying the DB2 Database

VisualAge Generator Developer provides a database profile that can be used to identify a specific database to be used during development and testing.

Several environment variables are also available to identify the database that to be used during development, run time, or both. These environment variables are used in this order, depending on target runtime environment (if the aim is development, the runtime environment variables do not apply unless generated applications are called by ITF):

FCWDBNAME_appl	VisualAge Generator environment variable. Used at run time for OS/2, Windows NT, and CICS NT environments. The value of <i>appl</i> identifies the application name that will use the database named in the environment variable setting.
ELARTRDB_tttt	VisualAge Generator environment variable. Used at runtime for CICS OS/2 environments. The value of <i>tttt</i> identifies the CICS transaction that will use the database named in the environment variable setting.
EZERSQLDB	VisualAge Generator environment variable. Used during development and at run time.
DB2DBDFT	DB2 environment variable. Used to identify the default database that will be used to support database access at: • Run time when no explicit database connection is issued and no environment variables are active • Development when there is no explicit database named in the database profile or active environment variables.

Note: For DB2 V1.2, the environment variable name is SQLDBDFT.

Working with Environment Variables

The value of an environment variable in OS/2 can be determined by issuing a set command with the name of the environment variable. The response is the current setting in the active OS/2 window, which is based on the CONFIG.SYS settings, but could have been modified later in the OS/2 window session.

For example, if you type: `set ezersqldb`

you see a response of: `EZERSQLDB=(null)`

If you want to set the environment variable, type `set ezersqldb=sample`

This is only the setting in the active OS/2 window. Global settings must be defined in the CONFIG.SYS file.

The value of an environment variable in Windows environments (Windows 95 or Windows NT) can be determined by issuing the set command without any parameters. The value of all environment variables then scrolls by in the active window. You cannot determine the setting of just one environment variable with the set command in Windows as you can in OS/2.

When a database is defined for a generated application using the available VisualAge Generator environment variables, the implicit database connection identifies the database being used. Subsequent database access uses this active database unless an alternative database is selected using the `EZECONCT` statement in the VisualAge Generator application logic.

Database connection activity is visible in the trace log (defaults to `FCWTRACE.OUT`) that is written when VisualAge Generator generated C++ applications are run with this environment variable setting: `SET FCWTRPOPT=31`

The active setting of the database environment variables used by VisualAge Generator generated C++ applications and the resulting use of an implicit database connection is visible in the trace logs shown in Figure 87 through Figure 88 on page 110.

C++ Generated Application Database Environment Variable Settings:

```
FCWDBNAME_VGL20S2=(null)
EZERSQLDB=current1
DB2DBDFT=(null)
```

FCWTRACE.OUT Log File:

```
(00290)<01:39:26>      -> CS0::CMINIT()      rc = 0
VGL20S2 (00290)<01:39:26>Using RSC name, fcw.rsc
VGL20S2 (00290)<01:39:26>Found EZERNLS environment variable, name=ENU
VGL20S2 (00290)<01:39:26> -> VGL20S2::CALLED
VGL20S2 (00290)<01:39:33>      -> ( SQL::DFTCONN ) Database = (current1) rc = 0
VGL20S2 (00290)<01:39:33>      User = () Password = () UOW = ( R )
VGL20S2 (00290)<01:39:33>      -> SRV_COMMON_MAIN
VGL20S2 (00290)<01:39:33>      -> STAFF-INQ
VGL20S2 (00290)<01:39:33>      -> ( SQL::INQUIRY ) Handle = 4
VGL20S2 (00290)<01:39:34>      rc = 0
```

Figure 87. Trace of Implicit EZERSQLDB Database Connection and Successful Database Access

C++ Generated Application Database Environment Variable Settings:

```
FCWDBNAME_VGL20S2=(null)
EZERSQLDB=(null)
DB2DBDFT=carrent1
```

FCWTRACE.OUT Log File:

```
(00308)<02:01:46>      -> CSO::CMINIT()      rc = 0
VGL20S2 (00308)<02:01:47>Using RSC name, fcw.rsc
VGL20S2 (00308)<02:01:47>Found EZERNLS environment variable, name=ENU
VGL20S2 (00308)<02:01:47> -> VGL20S2::CALLED
VGL20S2 (00308)<02:01:49>      -> ( SQL::DFTCONN ) Database = (           ) rc = 0
VGL20S2 (00308)<02:01:49>      User = ( ) Password = ( ) UOW = ( R )
VGL20S2 (00308)<02:01:49>      -> SRV_COMMON_MAIN
VGL20S2 (00308)<02:01:49>      -> STAFF-INQ
VGL20S2 (00308)<02:01:49>      -> ( SQL::INQUIRY ) Handle = 4
VGL20S2 (00308)<02:01:51>      rc = 0
```

Figure 88. Trace of Implicit DB2DBDFT Database Connection and Database Access.

The implicit database connection shown (where the DB2DBDFT environment variable was used to identify the active database name) looks the same as the log file shown in Figure 89 on page 110. The only difference is that the subsequent database access succeeds because a default database was identified.

C++ Generated Application Database Environment Variable Settings:

```
FCWDBNAME_VGL20S2=(null)
EZERSQLDB=(null)
DB2DBDFT=(null)
```

FCWTRACE.OUT Log File:

```
(00296)<01:41:21>      -> CSO::CMINIT()      rc = 0
VGL20S2 (00296)<01:41:21>Using RSC name, fcw.rsc
VGL20S2 (00296)<01:41:21>Found EZERNLS environment variable, name=ENU
VGL20S2 (00296)<01:41:21> -> VGL20S2::CALLED
VGL20S2 (00296)<01:41:22>      -> ( SQL::DFTCONN ) Database = (           ) rc = 0
VGL20S2 (00296)<01:41:22>      User = ( ) Password = ( ) UOW = ( R )
VGL20S2 (00296)<01:41:22>      -> SRV_COMMON_MAIN
VGL20S2 (00296)<01:41:22>      -> STAFF-INQ
VGL20S2 (00296)<01:41:22>      -> ( SQL::INQUIRY ) Handle = 4
VGL20S2 (00296)<01:41:22>      rc = -1024
```

Figure 89. Trace of Implicit Database Connection and Database Access Failure. When no database is defined for a generated application using the available environment variables, the implicit database connection will execute successfully, but any subsequent database access will fail.

Implicit database connections exist for all applications generated by VisualAge Generator that might use a database (that is, they have SQL statements in the application). An implicit database connection is always attempted during initialization of a generated application. This happens before programmer-defined logic takes control.

Notes:

1. The implicit database connection does not occur when testing an application using ITF. A database connection, based on the VisualAge Generator Developer database profile or the EZERSQLDB environment variable setting, is made only when an SQL statement is about to be processed. This restriction can affect the use of the EZECONCT statement in application logic. If you reset a database connection or query the current status of an existing connection, the response encountered during ITF may differ from that for runtime processing. This possibility should be factored into both application design and testing.
2. A possible change to make VisualAge Generator V2.2 work like V2.0 may be made by development staff. The refresh level (Fixpak 2) of VisualAge Generator V2.2 issues a request to start the database manager and to make the default database

connection for all calls to generated C++ applications. This did not occur in V2.0. A change to this processing logic may be made and released in a new Fixpak to modify the current behavior of V2.2 generated C++ applications.

5.2.2 Implementing Distributed Unit of Work Database Access

VisualAge Generator has always provided remote database access and support for database switching with remote unit-of-work support as implemented by the EZECONNECT statement. Remote unit-of-work support means that while a VisualAge Generator application can update two different databases, each database must be accessed and the update committed in a separate unit of work.

This section explains the database connection requirements, design issues, and describes a sample application that implements DUOW programming support. A full description of the DUOW sample application is provided in A.2, "DUOW Sample Application" on page 168.

During our study, we relied upon the volume *Distributed Relational Database Cross Platform Connectivity and Applications* for guidance on implementing database connections.

5.2.2.1 Using the EZECONCT Statement

The EZECONCT statement has been part of VisualAge Generator for several releases, but the support for DUOW processing is new.

With distributed unit-of-work support as provided by the DB2 product set and the enhancements to the EZECONNECT statement as provided by VisualAge Generator Version 2.2, one VisualAge Generator application can:

- Connect to multiple database within a single unit of work
- Have one active connection at any one time.

DUOW support as provided by the EZECONNECT statement is implemented using the CONNECT and DISCONNECT statements. Figure 90 on page 112 contains a review of the syntax of the EZECONCT statement.

```
CALL EZECONCT userid, pswd, servername, product, release, uow
```

uow is an eight character data item or literal with one of the following values:

R	Remote Unit of Work (default)
Dxy	Distributed Unit of Work
x	Syncpoint Option
1	One phase commit, single database update
2	Two phase commit, multiple database update
y	Disconnect Option
E	Explicit
C	Conditional
A	Automatic
DISC	Disconnect from named server database
DCURRENT	Disconnect from current database
DALL	Disconnect from all databases
SET	Reconnect to already connected database

Figure 90. VisualAge Generator EZECONCT Statement Implementation for DUOW Support

Full details for the syntax of the EZECONCT statement are available in the volume *Designing and Developing VisualAge Generator Applications* and the VisualAge Generator Developer help facility.

To quickly access the EZECONCT help topic:

1. Create a new process or statement group.
2. Use the insert icon (the arrow) or Cntl+I to trigger the statement template.
3. Select a statement type of CALL.
4. Click on the **EZE Words...** push button
5. Select the **EZECONCT** entry in the list of field EZE words.
6. Click on the **help** push button.

DUOW processing with the EZECONCT statement is supported during ITF processing and by generated C++ applications in these environments:

- OS/2
- Windows NT
- AIX
- CICS/NT
- CICS/6000

To implement DUOW processing with the EZECONCT statement, you can use these versions of DB2:

- DB2 V1.2 with this support for the UOW options:
 - UOW = R
 - UOW = DCURRENT (implemented as RESET)
 - UOW = DALL (implemented as RESET)
- DB2 V2.1 or later with support for all UOW options.

Implementation of three database functions is provided by VisualAge Generator client/server middleware processing:

- DB2 CONNECT
- SQL COMMIT WORK
- SQL ROLLBACK WORK

The middleware processing for the listed database functions is also used by ITF in VisualAge Generator V2.2. This allows for a coordinated commit between GUI and non-GUI applications running in an ITF environment and calls to local OS/2 generated C++ applications (as implemented by ITF when using a linkage table).

The coordinated commit between ITF and local OS/2-generated C++ applications is for all VisualAge Generator applications that access DB2 databases, not just those using DUOW processing.

When using both ITF and local OS/2-generated C++ applications, it is recommended that you process all EZECOMIT and/or EZEROLLB requests in applications being tested by ITF to ensure appropriate integrity for database updates.

DUOW processing is supported in CICS/6000 and CICS NT environments. Guidelines for database connection processing in these environments include these:

- Do not use EZESQLDB or FCWDBNAME_appl environment variables for default (implicit) database connection support if DB2 is an XA resource.
- For the best performance, have all transactions in the CICS region use the default database.
- If access to the nondefault database is required, code explicit database connects, in each transaction, with an automatic disconnect (DxA uow option, see Figure 90 on page 112) after a commit or rollback.

Additional details are provided in “Accessing Distributed Databases” in the volume *Designing and Developing VisualAge Generator Applications*.

5.2.2.2 Identifying a Transaction Manager Database

A valid DUOW database environment must exist before using VisualAge Generator to implement DUOW processing logic. In DB2 terms, this means that a connection between the client and target databases can be made with proper authorization and that a transaction manager database is configured as part of the DB2 environment.

On OS/2 with DB2/2, we used the Client Setup icon to start the configuration program. Using the **Client** and then **Configure...** menu options we started the client configuration dialog. The transaction manager database is defined on the Logging notebook page of the client configuration dialog. We tested our sample DUOW application using both available transaction manager database choices:

First Database Connected to

This sets the transaction manager database to the first database connected to, using a DUOW connection option.

Other

This sets the transaction manager database to the database identified in the transaction-manager database entry field.

The choice of a transaction manager database can affect database naming, and connection configuration as well as application programming rules. Note that the transaction manager database cannot be accessed using DRDA protocols.

Additional detail on the use of a transaction manager database is available in the volumes *DB2 Administration Guide* and *Distributed Relational Database Cross Platform Connectivity and Applications*.

5.2.2.3 Connecting to Multiple DB2 Databases

Basic database connection processing and authorization checking rules remain the same for DUOW processing. These rules were reviewed in 5.2.1.2, "Connecting to the DB2 Database" on page 107 and 5.2.1.3, "Identifying the Database Authorization ID" on page 107.

The database connect statement can include a user ID and password for the database logon to be used for authorization checking:

```
DB2 CONNECT TO DATABASE name USER userid USING password
```

This allows you to specify the authorization ID that will be used by DB2 instead of using a local or LAN logon identifier. This is important, as the target databases can require different authorization IDs and passwords. In some environment, the IDs and passwords can be case sensitive.

VisualAge Generator implements the EZECONCT statement using DB2 CONNECT TO DATABASE processing. When you use the VisualAge Generator EZECONCT statement to explicitly connect to a database, user ID and password values are passed as part of the request.

VisualAge Generator applications do not provide a user ID or password value for implicit database connections. Instead, the active local, LAN, or node logon is passed to the target database.

With explicit database connections implemented using the EZECONCT statement, there are two basic requirements:

1. You must have a valid local, LAN, or node logon to satisfy the implicit database connection performed by the VisualAge Generator application at startup (or when called).
If a valid logon is not available, a local logon prompt dialog is shown.
2. You must provide a user ID and password value that is valid for the target database.

This user ID or password data can be hard coded in the application EZECONCT statement or be derived from the active local, LAN, or node logon.

You can cancel the VisualAge Generator-triggered local logon prompt dialog requests and still get an application that uses EZECONCT statements to access a database successfully but this is not practical in an operational environment:

- If the VisualAge Generator application that accesses the database runs on the end-user's workstation, the end-user must cancel every logon request triggered by an implicit database connect
- If the VisualAge Generator application is a server running on a remote platform, the logic pauses to wait for the logon dialog to be completed. If this occurs on the screen of an unattended server workstation, the system hangs up.

This means that you need some form of active logon to support the implicit database connection that occurs in all VisualAge Generator applications that access a relational database—even those that use the EZECONCT statement to manage database connections with user ID or password parameters supplied by the application logic.

This means that to run a DUOW application you still need a valid logon, but you can avoid an actual database connection by not setting any of the environment variables that are used to identify the active relational database (see 5.2.1.4, "Identifying the DB2 Database" on page 108). A valid connection must be established before SQL processing can be performed.

5.2.2.4 Using ITF to Test DUOW Support

DUOW applications can be developed and tested using the ITF and local databases. DUOW means accessing multiple databases; it does not require that the databases be on multiple platforms.

If you implement more than one database in your local DB2/2 environment, you can issue EZECONCT statements to switch between them during ITF processing. The completed application, when generated, can access multiple local databases, a mix of local and remote databases, or multiple remote databases.

Before you use the EZECONCT statement in an application, you should test that you can issue DB2 CONNECT TO database statements from the command line in your OS/2 development environment. Once connected to the target database, with the appropriate authority, you should issue a DB2 SELECT ... statement to validate access authority.

By first using DB2 command line processing to test your database connections you reduce the complexity of resolving problems. The DB2 Database Director program is also useful for ensuring that connections and authorities are functioning appropriately.

There are several issues to consider when using ITF to test applications that use EZECONCT statements to manage database connections:

- There is no implicit database connection.
Applications running in ITF do not issue an implicit database connection when started. The implicit database connection is issued only when a generated application that could process SQL statements is started or called. This means that you cannot, using the EZECONCT statement, test for the current (implicit) connection in ITF, but you can test for it at run time.
- EZECONCT statement errors must be manually skipped under ITF.
When an SQL statement issued as part of a process option fails in ITF, processing continues and the EZESQLCD EZE word contains the error code. When an EZECONCT statement fails in ITF, the error code is loaded into the EZESQLCD EZE word, but a dialog window with additional error detail is also shown. Testing stops at this point. To continue, you must skip the statement (the error keeps occurring if you attempt to continue) but you can still check the value of the EZESQLCD EZE word to validate application error logic.
- Runtime database rules apply if you call generated applications.
If you use a linkage table to have ITF call generated versions of VisualAge Generator applications, you need to consider the database implications. The generated application will issue an implicit database connect. The database name used in the connect will be based on the environment variable settings used to control VisualAge Generator application database access (see 5.2.1.4, "Identifying the DB2 Database" on page 108).

It is possible to end up with ITF supporting access to a database name identified in the VisualAge Generator database preferences profile, while the generated applications called use a different database as identified by the environment variables.

When you use the EZECONCT statement in your logic, the database name is hardcoded. You may be forced to use one set of database names during testing and then another for actual production use. It may be helpful if the database names are parameters to the application or if a VisualAge Generator table is used to contain the database names and other information needed for using the EZECONCT statement.

Note: We used a VisualAge Generator table to hold database name, user ID, and password data in our DUOW sample application. Be careful with this approach. The user ID and password information is visible in the generated VisualAge

Generator table, which would represent a security exposure for a production system.

- Do not define a current ID for use as the table qualifier in the VisualAge Generator Developer database preferences profile.

Using a current ID value in the database preferences profile works fine when you are connecting to an MVS database and want to use unqualified table names in your VisualAge Generator SQL record definitions. If you are using a local database, the value is ignored for standard SQL activity.

The problem arises when you are using DUOW processing. There seems to be a bug because the first EZECONCT statement, whether a connect or a disconnect, triggers an SQL error code of -900. If the same logic is run a second time without stopping the test cycle, it succeeds.

5.2.3 Sample DUOW Application Configurations

To demonstrate the use of the EZECONCT statement to implement DUOW processing we have provided a sample application that implements both remote and distributed unit of work processing.

Note: The design guidelines in “Accessing Distributed Databases” in *Designing and Developing VisualAge Generator Applications* recommend that you not mix remote and distributed database connections. In our sample, we have done so only to provide support for independent testing of each target database and to show the differences in the use of the EZECONCT statement.

A full description of the DUOW sample application is provided in A.2, “DUOW Sample Application” on page 168. This section reviews the configuration of the DUOW sample application, including the remote database connections, for selected development and runtime environments.

5.2.3.1 Database Configuration

This book is not a guide to DB2 database configuration. The DB2 product manuals, supplemented by *Distributed Relational Database Cross Platform Connectivity and Applications*, SG24-4311, provide the required support for configuring a distributed database environment.

In this section, we quickly review the environment we used to validate the use of remote and distributed database access as implemented in the DUOW sample application. Configuration comments are made, but they may not be sufficient as a guide for you to complete a full distributed database configuration.

We used many configurations and hardware platforms during our study to test different methods of implementing database access.

DB2/2, DB2/6000, and DB2/MVS environments were connected using the connectivity functions of DB2 common server, and for the MVS target, DDCS/2.

5.2.3.2 DB2 Common Server Connections

The DB2 common server database environment we implemented is shown in Figure 91.

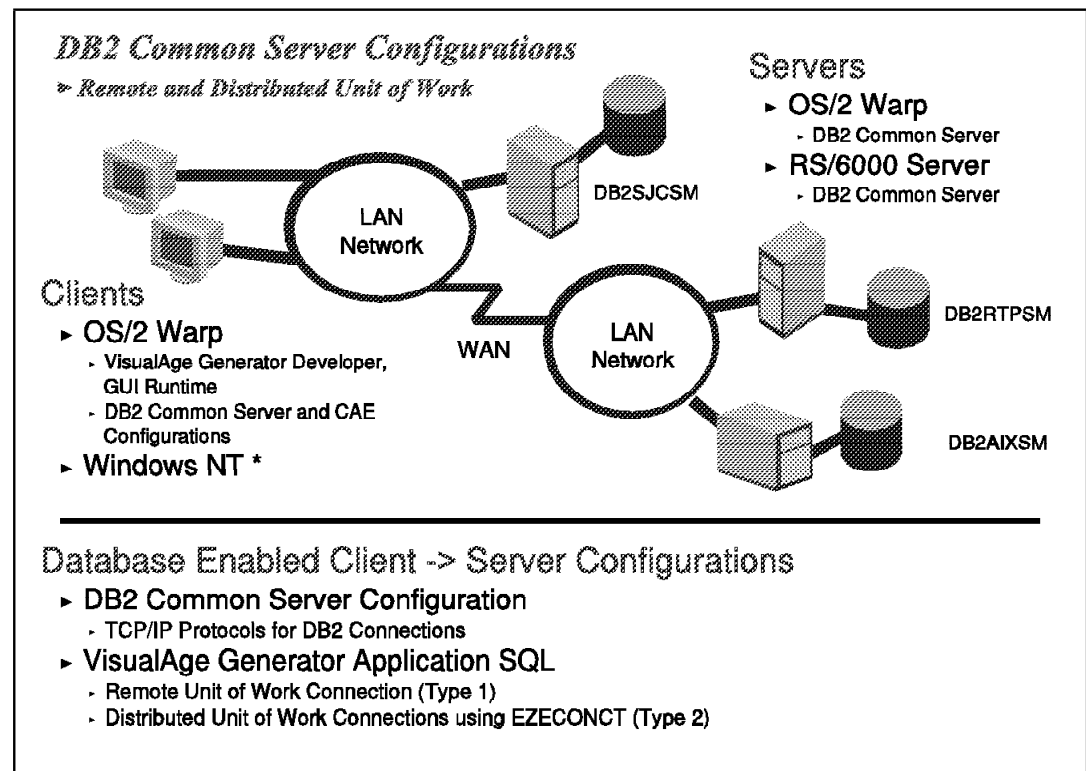


Figure 91. DB2/2 Common Server Configuration

We used TCP/IP as the protocol for DB2 common server connections. This was very simple to implement and provided sufficient function for our needs. You should evaluate the benefits and limitations of each of the available DB2 common server connection protocol options when designing your distributed database environment.

The key tasks in creating a TCP/IP-based database connection are these:

1. Enable TCP/IP support on the client and server workstations.

TCP/IP needs to be installed and configured. Each workstation should have a host name and you should be able to test connectivity between the hosts using the *ping* command. For example, *pinging* the Raleigh AIX system would look like:

```
[C:]ping vgrisc.raleigh.ibm.com
PING vgrisc.raleigh.ibm.com: 56 data bytes
64 bytes from 9.67.172.228: icmp_seq=0. time=313. ms
```

2. Teach DB2 to provide support for TCP/IP communications.

This is done using the DB2COMM environment variable. In OS/2, set the DB2COMM environment variable in the CONFIG.SYS file. This example shows a setting that supports Named Pipes and TCP/IP communication:

```
SET DB2COMM=NPIPE,TCPIP
```

For an AIX workstation, you need to update the DB2COMM environment variable in the db2profile file stored in the sqllib directory.

3. Define DB2 server service name (SVCENAME) for TCP/IP support.

This can be done using the DB2 database manager configuration dialog or DB2 commands. You can use these two commands to list the current database manager configuration and update the current database manager configuration setting for the DB2 server service name:

```
db2 get database manager configuration
db2 update database manager configuration using svcename db22sjcsrv
```

4. Add DB2 server service name to client and server TCP/IP services files.

Our client workstation can connect to several databases using DB2 common server TCP/IP based communication. The services file for our client workstation contains service name entries for each connected DB2 common server platform:

```
#      VGCS Database Connection Networking
#
db22sjcsrv      3600/tcp  # DB22 SJC Srvr Database System
db22sjcsrvi     3601/tcp  # DB22 SJC Srvr Database System Interrupt Port
db2instlc       3700/tcp  # DB2AIX RTP Srvr Database System
#db2instli      3701/tcp  # DB2AIX RTP Srvr Database System Interrupt Port
db22rtpsrv      3800/tcp  # DB22 RTP Srvr Database System
db22rtpsrvi     3801/tcp  # DB22 RTP Srvr Database System Interrupt Port
```

Note: The interrupt port services entry is required only when you need to support DB2 V1.x client connections.

5. Catalog database node on client workstation.

This tells the client workstation about the server workstation. This notification can be done using the Node Directory dialog available in the Database Director. To catalog the San Jose OS/2 DB2/2 server node on the client workstation, we used this command:

```
db2 catalog tcpip node db2sjc remote itscsrv1.almaden.ibm.com
server db22sjcsrv with 'DB2 on SJ VG Srvr'
```

6. Catalog the remote-node databases on the client workstation.

This gives the client workstation a database name to use for the remote database. The client database name does not have to be the same as the server database name. To catalog a database on the San Jose OS/2 DB2/2 server on the client workstation, we used this command:

```
db2 catalog database sample as db2sjcsm at node db2sjc
with 'DB2 Sample on OS/2 SJC Srvr'
```

7. Test the database connection

We can connect to a cataloged database using DB2 commands in a command window. A user ID and password can be provided as part of the connection. If they are not provided, DB2 uses an active node, local, or LAN logon, or prompt for a local logon (see 5.2.1.3, "Identifying the Database Authorization ID" on page 107).

The logon determines the creator ID used to resolve queries to unqualified table names.

Our current local logon is valid for the DB2SJC target environment so we can connect to the database and issue a select command from a command window using these commands:

```
[D:]db2 connect to db2sjcsm
```

Database Connection Information

```
Database product      = DB2/2 2.1.1
SQL authorization ID  = USERID
Local database alias  = DB2SJCSM
```

```
[D:\]db2 select * from staff where id=10
```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	DB2SJC	20	Mgr	7	18357.50	0.00

1 record(s) selected.

We have loaded the name field in our multiple instances of the staff table with a value that helps us know which database we have accessed (see 5.2.3.4, “Staff Tables” on page 123).

We repeated the process of cataloging DB2 common server nodes and databases based on the distributed database network shown in Figure 91 on page 117.

The node and cataloged database configuration, as seen from the Node Directory provided by the Database Director shown in Figure 92 along with System Database Directory detail views on the client workstation (HECATE).

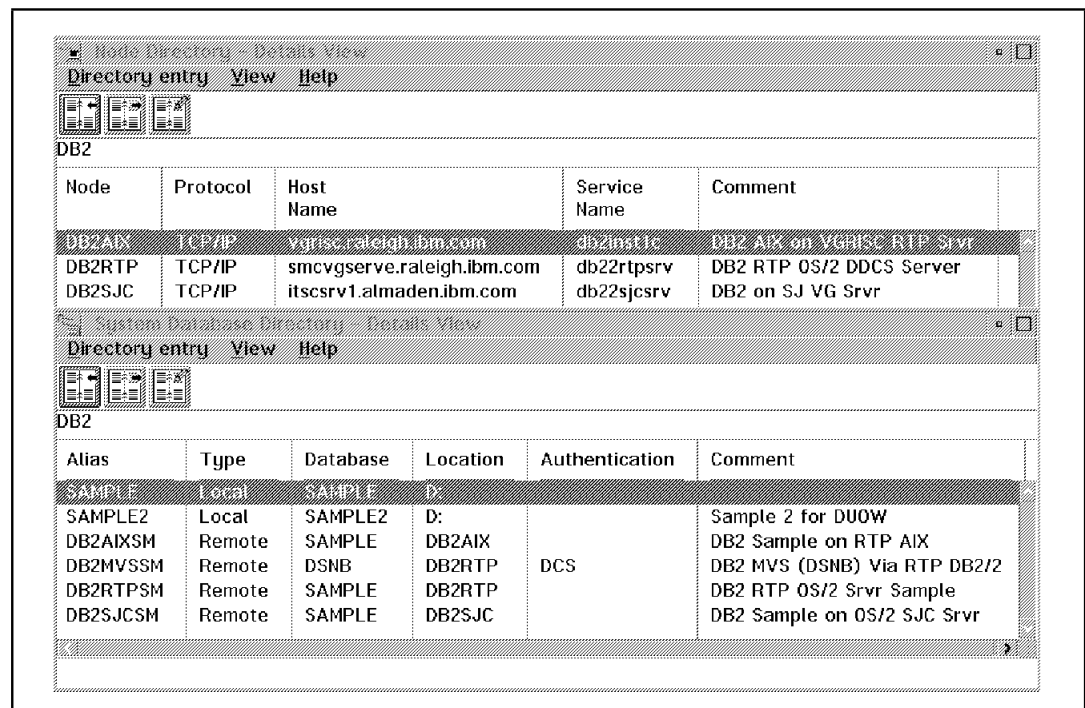


Figure 92. Distributed Database Nodes and Cataloged Databases. Both local and remote databases are shown in the System Database Directory detail view. We used the local databases for DUOW testing with VisualAge Generator Developer. You can also see an entry for an MVS database. Setting up a connection to an MVS database using DRDA is reviewed in 5.2.3.3, “DB2 DRDA Connections” on page 120.

5.2.3.3 DB2 DRDA Connections

To support access to a DB2/MVS V3.1 database environment, we connected a DB2/2 server to DB2/MVS using DRDA support. To implement the connection to the DB2/MVS database environment, we used DDCS/2 and Communications Manager/2 to implement a DRDA connection using APPC as the protocol option.

We then connected to the DB2/2 system using a DB2 client. This provided a shared gateway to DB2/MVS as shown in Figure 93.

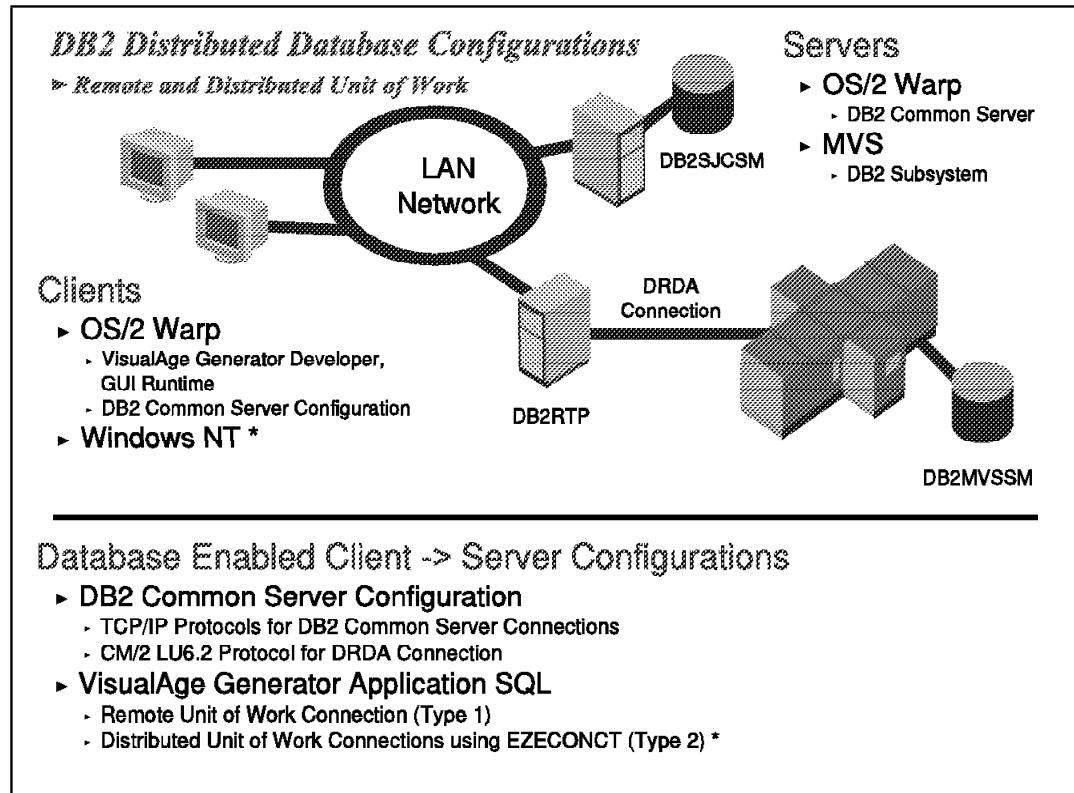


Figure 93. Remote DB2/2 and DB2/MVS Database Configuration

The key tasks in creating an APPC-based DRDA database connection are these:

1. Set up CM/2 session with DB2/MVS on DB2/2 server workstation
CM/2 configuration issues are as follows:
 - CM/2 V1.1 is a prerequisite but for two-phase-commit support CM/2 V.1.2 is required.
 - CM/2 installation must include APPC support,
 - Configuration includes an LU6.2 session between the local logical unit (LU) and the partner LU and common programming interface for communications side information for the connection.

Figure 94 on page 121 shows the CM/2 network definition file (NDF) used to support our DRDA connection to DB2/MVS.

```

DEFINE_LOCAL_CP  FQ_CP_NAME(USIBMNR.NRIM7N00 )
                  DESCRIPTION(Local Node)
                  CP_ALIAS(NRIG1I00)
                  NAU_ADDRESS(INDEPENDENT_LU)
                  NODE_TYPE(EN)
                  NODE_ID(X'05D00000')
                  NW_FP_SUPPORT(NONE)
                  HOST_FP_SUPPORT(YES)
                  HOST_FP_LINK_NAME(HOST0001)
                  MAX_COMP_LEVEL(NONE)
                  MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK  LINK_NAME(HOST0001)
                     DESCRIPTION(Connection to the USIBMNR Network)
                     FQ_ADJACENT_CP_NAME(USIBMNR.NRMCMC1 )
                     ADJACENT_NODE_TYPE(LEN)
                     DLC_NAME(IBMTRNET)
                     ADAPTER_NUMBER(0)
                     DESTINATION_ADDRESS(X'40002042045104')
                     ETHERNET_FORMAT(NO)
                     CP_CP_SESSION_SUPPORT(NO)
                     SOLICIT_SSCP_SESSION(YES)
                     ACTIVATE_AT_STARTUP(YES)
                     USE_PUNAME_AS_CPNAME(NO)
                     LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                     LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                     MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                     EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                     COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                     COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                     SECURITY(USE_ADAPTER_DEFINITION)
                     PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOCAL_LU  LU_NAME(NRRG1I00)
                 DESCRIPTION(Local LU)
                 LU_ALIAS(NRRG1I00)
                 NAU_ADDRESS(INDEPENDENT_LU);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(USIBMNR.NRARDSNB )
                  DESCRIPTION(DSNB on CARMVS1 - DB2 3.1)
                  PARTNER_LU_ALIAS(NRARDSNB)
                  PARTNER_LU_UNINTERPRETED_NAME(NRARDSNB)
                  MAX_MC_LL_SEND_SIZE(32767)
                  CONV_SECURITY_VERIFICATION(NO)
                  PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(USIBMNR.NRARDSNB )
                           DESCRIPTION(DSNB on CARMVS1 - DB2 3.1)
                           WILDCARD_ENTRY(NO)
                           FQ_OWNING_CP_NAME(USIBMNR.NRMCMC1 )
                           LOCAL_NODE_NN_SERVER(NO);

```

Figure 94 (Part 1 of 2). Communications Manager Network Definition File for DRDA Connection

```

DEFINE_MODE MODE_NAME(IBMDB2LM)
            DESCRIPTION(LU 6.2 Log Mode for DB2 on MVS)
            COS_NAME(#CONNECT)
            DEFAULT_RU_SIZE(NO)
            MAX_RU_SIZE_UPPER_BOUND(4096)
            RECEIVE_PACING_WINDOW(8)
            MAX_NEGOTIABLE_SESSION_LIMIT(32767)
            PLU_MODE_SESSION_LIMIT(4)
            MIN_CONWINNERS_SOURCE(2)
            COMPRESSION_NEED(PROHIBITED)
            PLU_SLU_COMPRESSION(NONE)
            SLU_PLU_COMPRESSION(NONE);

DEFINE_MODE MODE_NAME(QPCSUPP )
            COS_NAME(#CONNECT)
            DEFAULT_RU_SIZE(NO)
            MAX_RU_SIZE_UPPER_BOUND(1024)
            RECEIVE_PACING_WINDOW(7)
            MAX_NEGOTIABLE_SESSION_LIMIT(32767)
            PLU_MODE_SESSION_LIMIT(64)
            MIN_CONWINNERS_SOURCE(32)
            COMPRESSION_NEED(PROHIBITED)
            PLU_SLU_COMPRESSION(NONE)
            SLU_PLU_COMPRESSION(NONE);

DEFINE_DEFAULTS IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                DEFAULT_MODE_NAME(BLANK)
                MAX_MC_LL_SEND_SIZE(32767)
                DIRECTORY_FOR_INBOUND_ATTACHES(*)
                DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                DEFAULT_TP_CONV_SECURITY_RQD(NO)
                MAX_HELD_ALERTS(10);

DEFINE_CPIC_SIDE_INFO SYMBOLIC_DESTINATION_NAME(NRARDSNB)
                     DESCRIPTION(CPIC side info for DB2 v3.1 om CARMVS1)
                     PARTNER_LU_ALIAS(NRARDSNB )
                     MODE_NAME(IBMDB2LM)
                     SNA_SERVICE_TP_NAME(X'07',6DB);

```

Figure 94 (Part 2 of 2). Communications Manager Network Definition File for DRDA Connection

The CM/2 configuration process is reviewed in the manual *Installing and Using OS/2 Clients*. You may find the guidance available in *DB2 for MVS Connections with AIX and OS/2* helpful in when performing this task.

2. Set up DB2/2 Server and DDCS/2 as DB2/2 Gateway to DB2/MVS:
 - a. Define a remote database node for the DB2/MVS target database using APPC support. The Database Director Node Directory Entry catalog dialog or a command can be used to catalog the remote node. We used this command:

```
db2 catalog appc node NRARDSNB remote NRARDSNB
```

- b. Define a Distributed Connection Services (DCS) alias to be used for the remote MVS DB2 node. We used the DB2/MVS subsystem name (DSNB) as our DCS alias for the DB2/MVS node using this command:

```
db2 catalog dcs database DSNB as NRARDSNB
```

- c. Define the remote MVS DB2 subsystem as a database that is remote to DB2/2. The security authentication is directed to the DCS component.

During the following command, a node logon must be active with the user ID at the authority to perform this task:


```
db2 catalog database DSNB as DSNB at node NRARDSNB authentication dcs
```

d. Bind DDCS/2 to DB2/MVS

This allows DB2 commands to be processed against the target database that is accessed using DDCS/2. There are multiple list (.lst) files of package names that can be bound to target databases, as required, to enable client support for DB2 command processing. These are reviewed in detail in the manual *Installing and Using OS/2 Clients*

During the following command, a node logon must be active with a user ID that has the required bind authority:

```
db2 bind D:SQLLIBND@ddcmvs.lst grant public
```

3. Set up DB2/2 Client connection to DB2/2-DDCS/2 Gateway to DB2/MVS.

The UPM node logon ID is used to access the remote database. This user ID must have the appropriate authorities in the target database environment. If the CL1 has no node logon active, then first try to access with another user ID. If a local or LAN logon is active, and this user ID does not have the right authority to access the remote resource and an error occurred (see common problems):

```
db2 catalog database DSNB as DB2MVSSM at node DB2RTP
```

You may wish to automate the node logon to provide easier support for remote database connections. This command could be added to your startup.cmd file to automate the node logon:

```
logon MVSID /P=secret /N=DB2RTP
```

You may have other local or LAN logons that are also required to support access to local databases or to shared resources.

5.2.3.4 Staff Tables

The DUOW sample application uses the DB2 sample database table named STAFF. The STAFF table is implemented in each of our cataloged databases at least once. Figure 95 on page 124 contains a review of the STAFF table as implemented in each of our cataloged databases.

Database Connection Information
 Database product = DB2/2 2.1.1
 SQL authorization ID = USERID
 Local database alias = SAMPLE

TBCREATOR	TBNAME	NAME	COLTYPE	LENGTH	SCALE	COLNO
USERID	STAFF	ID	SMALLINT	2	0	0
USERID	STAFF	NAME	VARCHAR	9	0	1
USERID	STAFF	DEPT	SMALLINT	2	0	2
USERID	STAFF	JOB	CHAR	5	0	3
USERID	STAFF	YEARS	SMALLINT	2	0	4
USERID	STAFF	SALARY	DECIMAL	7	2	5
USERID	STAFF	COMM	DECIMAL	7	2	6

7 record(s) selected.

Database Connection Information
 Database product = DB2/2 2.1.1
 SQL authorization ID = USERID
 Local database alias = SAMPLE2

TBCREATOR	TBNAME	NAME	COLTYPE	LENGTH	SCALE	COLNO
USERID	STAFF	ID	SMALLINT	2	0	0
USERID	STAFF	NAME	VARCHAR	9	0	1
USERID	STAFF	DEPT	SMALLINT	2	0	2
USERID	STAFF	JOB	CHAR	5	0	3
USERID	STAFF	YEARS	SMALLINT	2	0	4
USERID	STAFF	SALARY	DECIMAL	7	2	5
USERID	STAFF	COMM	DECIMAL	7	2	6

7 record(s) selected.

Database Connection Information
 Database product = DB2/2 2.1.1
 SQL authorization ID = USERID
 Local database alias = DB2SJCSM

TBCREATOR	TBNAME	NAME	COLTYPE	LENGTH	SCALE	COLNO
USERID	STAFF	ID	SMALLINT	2	0	0
USERID	STAFF	NAME	VARCHAR	9	0	1
USERID	STAFF	DEPT	SMALLINT	2	0	2
USERID	STAFF	JOB	CHAR	5	0	3
USERID	STAFF	YEARS	SMALLINT	2	0	4
USERID	STAFF	SALARY	DECIMAL	7	2	5
USERID	STAFF	COMM	DECIMAL	7	2	6

7 record(s) selected.

Figure 95 (Part 1 of 2). Staff Table Definitions for Distributed Database Environment

Database Connection Information

Database product = DB2/2 2.1.1
 SQL authorization ID = USERID
 Local database alias = DB2RTPSM

TBCREATOR	TBNAME	NAME	COLTYPE	LENGTH	SCALE	COLNO
USERID	STAFF	ID	SMALLINT	2	0	0
USERID	STAFF	NAME	VARCHAR	9	0	1
USERID	STAFF	DEPT	SMALLINT	2	0	2
USERID	STAFF	JOB	CHAR	5	0	3
USERID	STAFF	YEARS	SMALLINT	2	0	4
USERID	STAFF	SALARY	DECIMAL	7	2	5
USERID	STAFF	COMM	DECIMAL	7	2	6

7 record(s) selected.

Database Connection Information

Database product = DB2 MVS 3.1.0
 SQL authorization ID = STLCSP2
 Local database alias = DB2MVSSM

TBCREATOR	TBNAME	NAME	COLTYPE	LENGTH	SCALE	COLNO
STLCSP2	STAFF	ID	SMALLINT	2	0	1
STLCSP2	STAFF	NAME	VARCHAR	9	0	2
STLCSP2	STAFF	DEPT	SMALLINT	2	0	3
STLCSP2	STAFF	JOB	CHAR	5	0	4
STLCSP2	STAFF	YEARS	SMALLINT	2	0	5
STLCSP2	STAFF	SALARY	DECIMAL	7	2	6
STLCSP2	STAFF	COMM	DECIMAL	7	2	7

7 record(s) selected.

Database Connection Information

Database product = DB2/6000 2.1.0
 SQL authorization ID = DB2
 Local database alias = DB2AIXSM

TBCREATOR	TBNAME	NAME	COLTYPE	LENGTH	SCALE	COLNO
DB2	STAFF	ID	SMALLINT	2	0	0
DB2	STAFF	NAME	VARCHAR	9	0	1
DB2	STAFF	DEPT	SMALLINT	2	0	2
DB2	STAFF	JOB	CHAR	5	0	3
DB2	STAFF	YEARS	SMALLINT	2	0	4
DB2	STAFF	SALARY	DECIMAL	7	2	5
DB2	STAFF	COMM	DECIMAL	7	2	6

7 record(s) selected.

Figure 95 (Part 2 of 2). Staff Table Definitions for Distributed Database Environment

The staff table can be implemented in workstation environments using either the VisualAge Generator provided d:\ezerver2\samples\staff.IXF database export file or the DB2SAMPLE command provided with DB2/2 V2.1. The sample database can also be implemented on a host using the required utility commands.

To make testing easier, we loaded a specific record in each STAFF table instance, to confirm which target database had been accessed during DUOW sample application processing

5.2.3.5 Preparing to Test DUOW Applications

Before testing our DUOW sample application, we generated and prepared the GUI and called application components. We used a single system for the GUI and called application so that the linkage table we used for generation and runtime control requests support for a local call (see Figure 96).

```
:CALLLINK APPLNAME=DUOWAPP LIBRARY=DUOWAPP PARMFORM=oslink  
LINKTYPE=dynamic LUWCONTROL=server.
```

Figure 96. Linkage Table for DUOW Called Application

Figure 97 contains the generation commands we used to generate the DUOW sample application GUI and called application for use in an OS/2 environment.

```
EZE2GEN GENERATE DUOWGUI /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%%EZEENV%%  
/linkage=h:\vgcs\applst.lkg /system=OS2GUI >e:\vgcs\genout\duowapp.log  
  
EZE2GEN GENERATE DUOWAPP /MSL=VGPIGZ /GENOUT=e:\vgcs\genout\%%EZEENV%%  
/linkage=h:\vgcs\applst.lkg /system=OS2 >e:\vgcs\genout\duowgui.log
```

Figure 97. Generate Statements for DUOW Sample Application. The default VisualAge Generator options /GENTABLES and /GENEMBEDDEDGUIs ensured that when we generated DUOWGUI, the VisualAge Generator table and embedded GUIs were also generated.

During preparation, our DUOWAPP application was bound against only one database. Before we run the application against other database targets we have to bind the application once for each database. We use this bind statement when binding against local or remote database targets:

```
sqlbind duowapp.bnd <databasename>
```

Figure 98 shows the basic database activity for the DUOW sample application. (The DUOW sample application is reviewed in detail in A.2, "DUOW Sample Application" on page 168.)

```
Implicit DB2 Connect (Only for generated C++ applications, not ITF)  
  
If DB2 Disconnect Requested  
-> Issue DB2 Disconnect  
End  
  
If Single Database Access Requested (Remote Database Access)  
-> Connect to Target Database - Type R  
-> Process SQL Request  
-> Commit (Still connected to database)  
  
Else Double Database Access Requested (Distributed Unit of Work Database Access)  
-> Connect to Target Database 1 - Type D2A (Disconnect after commit)  
-> Process SQL Request  
-> Connect to Target Database 1 - Type D2A (Disconnect after commit)  
-> Process SQL Request  
-> Commit (Triggers disconnect)  
End  
  
Exit
```

Figure 98. DUOW Sample Application Database Connection Processing Logic

The standard testing cycle that was used for each DUOW configuration is shown in Figure 99 on page 127.

This test was performed once with and once without a database disconnect request:

- Test a cycle of single and double database access requests for select processing. The cycle ran two times. Values were provided for the EZECONCT statement user ID and password parameters.

These tests were performed twice, once with and once without values for the EZECONCT statement user ID and password parameters:

- Test single database access against each target database for select and update processing.
- Test double database access for the two target databases for select and update processing.

Figure 99. Standard Testing Cycle for DUOW Configurations

The approach in Figure 99 ensured we tested the basic EZECONCT statement functions, included update processing, and understood any issues related to using the EZECONCT statement to implement DUOW processing. Any problems related to mixing remote and distributed database connections in a single application could also be determined.

5.2.3.6 DUOW Using Local DB2/2 Databases

This first scenario represents a possible development and unit testing environment. We have two local databases, SAMPLE and SAMPLE2, which are cataloged on the client workstation. (See Figure 92 on page 119 for a list of all the local and remote databases cataloged on our client workstation.) The STAFF table exists in all our target databases.

Our test environment setup included:

1. A local logon to provide the required database authorization. We performed a local logon to UPM with this command:

```
logon userid /p=password /1
```

2. Starting the DB2/2 database environment with this command:

```
db2start
```

3. Database connection and selected processing tests to ensure that we could access the STAFF table in each target database. The connection and test SELECT statements for the SAMPLE and SAMPLE2 databases are shown in Figure 100 on page 128.

```
[H:\vgcs]db2 connect to sample

Database Connection Information
Database product      = DB2/2 2.1.1
SQL authorization ID  = USERID
Local database alias  = SAMPLE

[H:\vgcs]db2 select * from staff where id=10
```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	Sample	20	Mgr	7	18357.50	0.00

```
1 record(s) selected.

[H:\vgcs]db2 connect to sample2

Database Connection Information
Database product      = DB2/2 2.1.1
SQL authorization ID  = USERID
Local database alias  = SAMPLE2

[H:\vgcs]db2 select * from staff where id=10
```

COMM	DEPT	ID	JOB	NAME	SALARY	YEARS
0	20	10	Mgr	Sample2	18357.50	7.00

```
1 record(s) selected.
```

Figure 100. Local Database Connection and Test SELECT Statements

To prepare for running a generated version of the DUOW sample application, we bound the DUOWAPP application to the SAMPLE and SAMPLE2 databases using these commands:

```
sqlbind duowapp.bnd sample
sqlbind duowapp.bnd sample2
```

The active local logon ID (USERID) had the required DB2 authority to bind programs to the target databases.

The DUOW sample application can now be tested in an ITF or native OS/2 runtime environment. We ran the standard test set (Figure 99 on page 127) using both of these runtime environments. We used different environment variable and database-preference profile settings:

- No database name was defined using either the EZERSQLDB environment variable or the database preferences profile.
- The database name was defined using EZERSQLDB.
- The database name was defined using database preferences profile.

Figure 101 on page 129 shows a portion of the FCWTRACE.OUT file for a test run where user ID and password data was not used as part of the EZECONCT statement.

```

(00952)<10:28:10>      -> CSO::CMINIT()      rc = 0
DUOWAPP (00952)<10:28:10>Using RSC name, fcw.rsc
DUOWAPP (00952)<10:28:10>Found EZERNLS environment variable, name=ENU
DUOWAPP (00952)<10:28:10> -> DUOWAPP::CALLED
DUOWAPP (00952)<10:28:11>      -> ( SQL::DFTCONN ) Database = (          ) rc = 0
DUOWAPP (00952)<10:28:11>      User = ( ) Password = ( ) UOW = ( R )
DUOWAPP (00952)<10:28:11>      -> DUOW-MAIN
DUOWAPP (00952)<10:28:11>      -> DUOW-DB-DISCONNECT
DUOWAPP (00952)<10:28:11>      -> ( SQL::EZECONN ) Database = (          ) rc = 0
DUOWAPP (00952)<10:28:12>      User = ( ) Password = ( ) UOW = ( DALL )
DUOWAPP (00952)<10:28:12>      <- DUOW-DB-DISCONNECT
DUOWAPP (00952)<10:28:12>      -> DUOW-DB1
DUOWAPP (00952)<10:28:12>      -> DUOW-DB-CONNECT
DUOWAPP (00952)<10:28:13>      -> ( SQL::EZECONN ) Database = (SAMPLE          ) rc = 0
DUOWAPP (00952)<10:28:13>      User = ( ) Password = ( ) UOW = ( D2A )
DUOWAPP (00952)<10:28:13>      <- DUOW-DB-CONNECT
DUOWAPP (00952)<10:28:13>      -> DUOW-DB-ACCESS
DUOWAPP (00952)<10:28:13>      -> DUOW-STAFF-GET
DUOWAPP (00952)<10:28:13>      -> ( SQL::INQUIRY ) Handle = 9
DUOWAPP (00952)<10:28:13>      rc = 0
DUOWAPP (00952)<10:28:13>      -> ( SQL::CLOSE ) Handle = 9
DUOWAPP (00952)<10:28:13>      rc = 0
DUOWAPP (00952)<10:28:13>      <- DUOW-STAFF-GET
DUOWAPP (00952)<10:28:13>      <- DUOW-DB-ACCESS
DUOWAPP (00952)<10:28:13>      <- DUOW-DB1
DUOWAPP (00952)<10:28:13>      -> DUOW-DB2
DUOWAPP (00952)<10:28:13>      -> DUOW-DB-CONNECT
DUOWAPP (00952)<10:28:15>      -> ( SQL::EZECONN ) Database = (SAMPLE2          ) rc = 0
DUOWAPP (00952)<10:28:15>      User = ( ) Password = ( ) UOW = ( D2A )
DUOWAPP (00952)<10:28:15>      <- DUOW-DB-CONNECT
DUOWAPP (00952)<10:28:15>      -> DUOW-DB-ACCESS
DUOWAPP (00952)<10:28:15>      -> DUOW-STAFF-GET
DUOWAPP (00952)<10:28:15>      -> ( SQL::INQUIRY ) Handle = 9
DUOWAPP (00952)<10:28:16>      rc = 0
DUOWAPP (00952)<10:28:16>      -> ( SQL::CLOSE ) Handle = 9
DUOWAPP (00952)<10:28:16>      rc = 0
DUOWAPP (00952)<10:28:16>      <- DUOW-STAFF-GET
DUOWAPP (00952)<10:28:16>      <- DUOW-DB-ACCESS
DUOWAPP (00952)<10:28:16>      <- DUOW-DB2
DUOWAPP (00952)<10:28:16>      -> ( SQL::COMMIT ) rc = 0
DUOWAPP (00952)<10:28:16>      <- DUOW-MAIN
DUOWAPP (00952)<10:28:16> <- DUOWAPP::CALLED
DUOWAPP (00952)<10:28:16>      -> CSO::CMCLOSE() rc = 0

```

Figure 101. FCWTRACE.OUT File for Distributed Access to Local Databases

We learned the following:

Local logon required:

You need to be logged on locally to run the application in either the ITF or the native OS/2 runtime environment. The primary reason is that you must be logged on to start DB2/2.

Type R connections may require a disconnect:

After making a Type R connection to a single database successfully we tried to access both databases with a Type D2A connection. An SQL error message was received when we ran the cycle of tests without the database-disconnect request option in both the ITF and native OS/2 runtime environments:

SQL1246N

Connection settings cannot be changed while connections exist.

The Type R connection still existed—it had not been reset. When a Type R connection request follows a Type D2A, as performed in our DUOW sample application, there is no need to disconnect. The Type D2A connection forces an automatic disconnect after an SQL commit or rollback.

Implicit database connections may require a disconnect:

If we ran the DUOW sample application in native OS/2 environment and defined a database for use by the called application using either the EZERSQLDB or FCWDBNAME_appl environment variables, we had to disconnect from the VisualAge Generator implicit database first or we would receive the SQL1246N error message. Because we had identified a default database name, VisualAge Generator processed a default connection (see 5.2.1.4, "Identifying the DB2 Database" on page 108), to issue the Type D2A connections we had to disconnect from the implicit database. If we set the DB2DBDFT environment variable, we did not get an implicit connection that needed to be reset.

If we had never issued a Type R connection request using the EZECONCT statement and did not have a database defined using the EZERSQLDB or FCWDBNAME_appl environment variables, we never had to disconnect from an active database prior to issuing Type D2A connections.

Connections can use local logon information:

If we configured the DUOW sample application so that only a database name, without user ID or password values, was used in the EZECONCT statement, the application would succeed as long as the local logon was authorized to perform the connection and SQL request.

An SQL error was received when the local logon was not authorized to perform the SQL requests:

SQL0551N

<authorization-ID> does not have the privilege to perform operation
<operation> on object <name>.

This tells us we can write applications that do not use hard-coded user ID and password information in the EZECONCT statements if we ensure that a user logon is available for use as the database authorization ID (see 5.2.1.3, "Identifying the Database Authorization ID" on page 107 for more details).

If we did not have an active local or database node logon, we could still run the generated application, and it would work, but we had to constantly cancel local logon request dialogs.

Even though the DUOW sample application was configured to use hard-coded user ID and password values for the SQL connect processing, the use of SQL in the application triggered a logon request. Canceling the logon dialog did not affect the DUOW sample application processing.

Connections can use private logon information:

If we configured the DUOW sample application so that the database name, user ID, and password values were used in the EZECONCT statement, the application would succeed even when the local logon was not authorized to perform the connection and SQL request.

The local logon meant that we did not see logon dialogs, but the user ID and password passed on the EZECONCT statement was used as the database authorization ID.

So far, using just local databases, we see that VisualAge Generator applications that use the EZECONCT statement can be written to use either existing local logons or hard-coded user ID and password data for database authorization checking.

The rules are the same as when using the DB2 command line-processor interface. You can issue DB2 CONNECT TO database_name requests which will use the local (or node) logon. If not available, a logon request dialog is shown. You can also pass the required user ID and password data as part of the DB2 command line processor request:

```
DB2 CONNECT TO database_name USER userid USING password
```


This means that the first thing you should do before testing a remote or distributed database application is to prove that you can access the target databases and tables using the DB2 command line processor interface.

5.2.3.7 DUOW Using Local and Remote Databases

The distributed database configuration we built (see 5.2.3.1, “Database Configuration” on page 116) supports multiple DUOW configurations. We tested many of them. In this section we review a configuration that uses a local and a remote database (see Figure 91 on page 117).

To prepare to run the generated version of the DUOW sample application in the configuration shown in Figure 91 on page 117 we bound the DUOWAPP application to the DB2AIXSM database (we had already bound to the SAMPLE database) using this command:

```
sqlbind duowapp.bnd db2aixsm
```

The active local logon ID (USERID) did not have the required DB2 authority to bind programs to this remote target database. We were prompted to log on to the remote database node (DB2AIX, see Figure 92 on page 119); once the log on was completed, the bind finished successfully.

We now have a local and a remote node logon active on the system.

The DUOW sample application can now be tested in an ITF or native OS/2 runtime environment. We ran the standard test set (Figure 99 on page 127) for the native OS/2 runtime environment; a reduced test set was used to prove that the ITF could perform all required tasks.

We used these two authorization scenarios:

- Local and node logons are used to establish database authorization for the target databases. The user ID and password values used in the EZECONCT statement parameters are left blank.
- No logon exists for any of the target databases. User ID and password values are provided in the EZECONCT statement parameters to identify database authorization.

Figure 102 on page 132 shows a portion of the FCWTRACE.OUT file for a test run where user ID and password data was passed as part of the EZECONCT statement.

```

(00952)<10:36:47>      -> CSO::CMINIT()    rc = 0
DUOWAPP (00952)<10:36:48>Using RSC name, fcw.rsc
DUOWAPP (00952)<10:36:48>Found EZERNLS environment variable, name=ENU
DUOWAPP (00952)<10:36:48> -> DUOWAPP::CALLED
DUOWAPP (00952)<10:36:48>      -> ( SQL::DFTCONN ) Database = (          ) rc = 0
DUOWAPP (00952)<10:36:48>      User = ( ) Password = ( ) UOW = ( R )
DUOWAPP (00952)<10:36:48>      -> DUOW-MAIN
DUOWAPP (00952)<10:36:48>      -> DUOW-DB-DISCONNECT
DUOWAPP (00952)<10:36:49>      -> ( SQL::EZECONN ) Database = (          ) rc = 0
DUOWAPP (00952)<10:36:49>      User = ( ) Password = ( ) UOW = ( DALL )
DUOWAPP (00952)<10:36:49>      <- DUOW-DB-DISCONNECT
DUOWAPP (00952)<10:36:49>      -> DUOW-DB1
DUOWAPP (00952)<10:36:49>      -> DUOW-DB-CONNECT
DUOWAPP (00952)<10:36:50>      -> ( SQL::EZECONN ) Database = (SAMPLE          ) rc = 0
DUOWAPP (00952)<10:36:50>      User = (userid ) Password = (password) UOW = ( D2A )
DUOWAPP (00952)<10:36:51>      <- DUOW-DB-CONNECT
DUOWAPP (00952)<10:36:51>      -> DUOW-DB-ACCESS
DUOWAPP (00952)<10:36:51>      -> DUOW-STAFF-GET
DUOWAPP (00952)<10:36:51>      -> ( SQL::INQUIRY ) Handle = 9
DUOWAPP (00952)<10:36:51>      rc = 0
DUOWAPP (00952)<10:36:51>      -> ( SQL::CLOSE ) Handle = 9
DUOWAPP (00952)<10:36:51>      rc = 0
DUOWAPP (00952)<10:36:51>      <- DUOW-STAFF-GET
DUOWAPP (00952)<10:36:51>      <- DUOW-DB-ACCESS
DUOWAPP (00952)<10:36:51>      <- DUOW-DB1
DUOWAPP (00952)<10:36:51>      -> DUOW-DB2
DUOWAPP (00952)<10:36:51>      -> DUOW-DB-CONNECT
DUOWAPP (00952)<10:36:53>      -> ( SQL::EZECONN ) Database = (DB2AIXSM          ) rc = 0
DUOWAPP (00952)<10:36:53>      User = (db2 ) Password = (DB2 ) UOW = ( D2A )
DUOWAPP (00952)<10:36:53>      <- DUOW-DB-CONNECT
DUOWAPP (00952)<10:36:53>      -> DUOW-DB-ACCESS
DUOWAPP (00952)<10:36:53>      -> DUOW-STAFF-GET
DUOWAPP (00952)<10:36:53>      -> ( SQL::INQUIRY ) Handle = 9
DUOWAPP (00952)<10:36:54>      rc = 0
DUOWAPP (00952)<10:36:54>      -> ( SQL::CLOSE ) Handle = 9
DUOWAPP (00952)<10:36:55>      rc = 0
DUOWAPP (00952)<10:36:55>      <- DUOW-STAFF-GET
DUOWAPP (00952)<10:36:55>      <- DUOW-DB-ACCESS
DUOWAPP (00952)<10:36:55>      <- DUOW-DB2
DUOWAPP (00952)<10:36:56>      -> ( SQL::COMMIT ) rc = 0
DUOWAPP (00952)<10:36:56>      <- DUOW-MAIN
DUOWAPP (00952)<10:36:56> <- DUOWAPP::CALLED
DUOWAPP (00952)<10:36:56>      -> CSO::CMCLOSE() rc = 0

```

Figure 102. FCWTRACE.OUT File for Distributed Access to Local and Remote Databases

We learned the following:

Local logon if required:

Local logon is required for implicit database connection, database disconnect, and the actual remote database connection even when a remote node logon exists. If these logon prompts are canceled, the remote database access will succeed. The local logon prompt will reappear for each subsequent server call until a local logon is established.

DB2/6000 user ID or password is case sensitive:

Database user ID and password are case sensitive in the EZECONCT statement when the target environment, in our example, the RS/6000 with DB2/6000, is case sensitive.

Node logon is prompted if required:

If user ID and password data is not passed as part of the EZECONCT statement for a remote database, a node logon dialog is shown. This logon dialog does not seem to be case sensitive. The user ID is forced to upper case in the dialog, even though the EZECONCT statement must use a lower case user ID value to succeed (see Figure 102). The password, when entered in either lower or upper case, is accepted.

Disconnect is required after Type R connect:

Database connection issues and the possible occurrence of an SQL1246N error remained the same (see 5.2.3.6, "DUOW Using Local DB2/2 Databases" on page 127).

5.2.3.8 DUOW Using Remote Databases

In this section, we review a configuration that supports access to two remote databases. We tested three different configurations with only remote database access:

- Two remote DB2/2 databases
- Remote DB2/2 and DB2/6000 databases using DB2 CAE
- Remote DB2/2 and DB2/MVS databases

The first configuration used two remote DB2/2 databases where each was on a separate node (see Figure 103). We used the same client platform as before where we had a full DB2/2 installed with client and local database support.

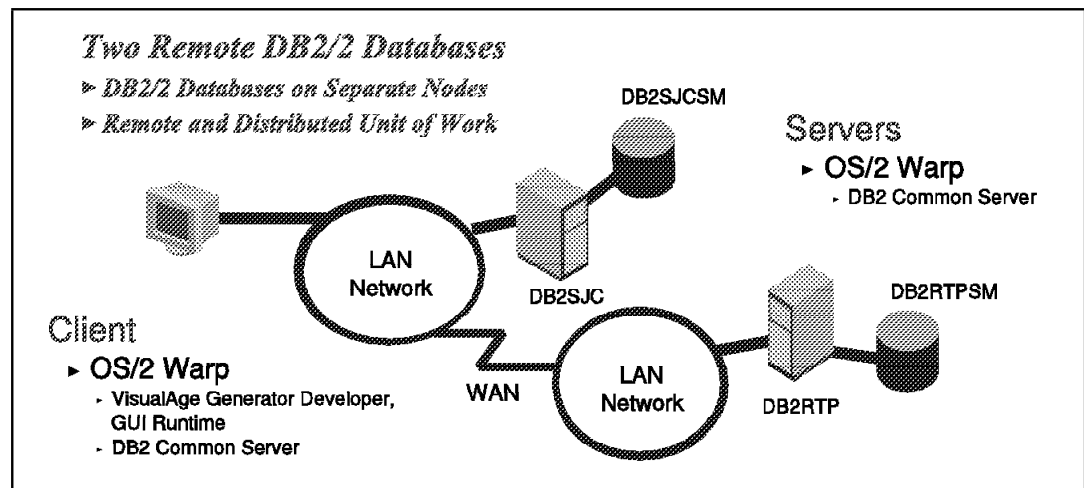


Figure 103. DUOW Configuration with Two Remote DB2/2 Databases on Different Nodes

To prepare to run the generated version of the DUOW sample application in the configuration shown in Figure 103, we bound the DUOWAPP application to the DB2SJC and DB2RTPSM databases using these commands:

```
sqlbind duowapp.bnd db2sjcsm
sqlbind duowapp.bnd db2rtpsm
```

The active local logon ID (USERID) had the required DB2 authority to bind programs to the target database on both platforms because:

- Our logon ID, USERID, was a valid administration ID on both platforms.
- The password was the same on both platforms.

The generated version of the DUOW sample application can now be tested for this configuration.

We ran the standard test set (Figure 99 on page 127) with the generated code. Our testing focused on these issues:

- Local and node logon requirements
- VisualAge Generator generated application DB2/2 control logic

We learned the following:

- A local logon is required to access the remote databases if the user ID and password values are used in the EZECONCT statement parameters.
Because we were using a client workstation that had DB2/2 installed with client and local database support, the local logon was required.
- While DB2/2 does not have to be started on the local workstation to access remote databases, it must be started there if you use a VisualAge Generator application to access the remote database.
The called VisualAge Generator application starts DB2/2 for you if it is there to be started. VisualAge Generator does not know whether you intend to use local databases, so DB2/2 is started regardless of your intent to process SQL statements.¹⁸

The second configuration used remote DB2/2 and DB2/6000 databases where each was on a separate node (see Figure 104). We used a client platform where we had DB2/2 installed with these features:

- Client Application Enabler
- Administrator's Toolkit

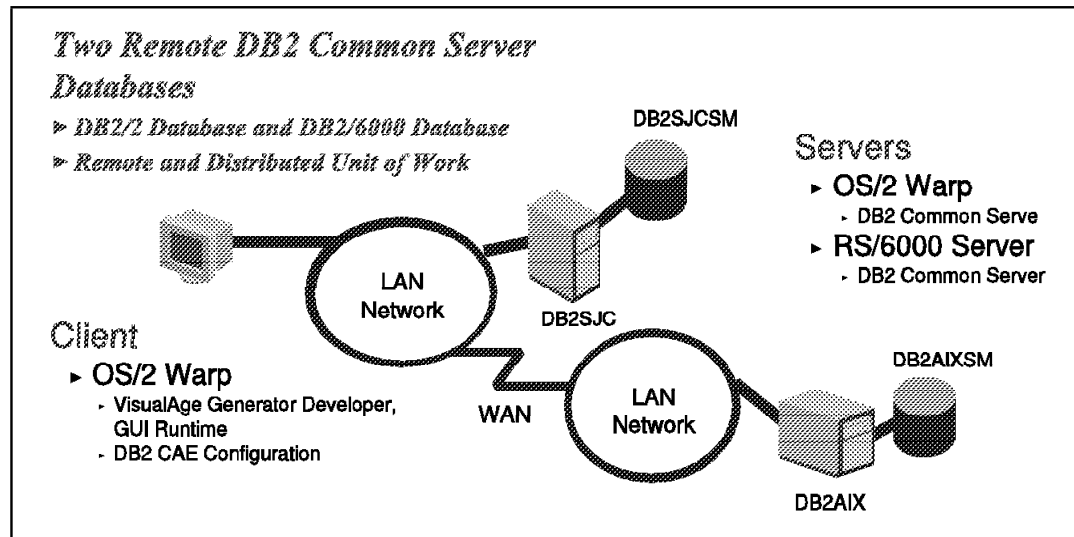


Figure 104. DUOW Configuration with Remote DB2/2 and Remote DB2/6000 Databases on Different Nodes

The DUOW sample application can now be tested in an ITF or native OS/2 runtime environment. We ran subsets of the standard test set (see Figure 99 on page 127) using both of these runtime environments. Our tests focused on:

- The use of databases with different environment variable and preferences profile settings
- Local and node logon requirements

We learned the following:

¹⁸ VisualAge Generator generated applications seem to issue a start database request when first called and a database connection request (we have termed this the *implicit database connection*) for each call. This imposes overhead that, currently, cannot be avoided. The best you can do is not define a default database, using the EZESQLDB or FCWDBNAME_appl environment variables, when you will be using EZECONCT statements to control database access for **all** your SQL processing. The IBM VisualAge Generator development lab is considering alternatives that may reduce the initial cost of calling VisualAge Generator applications. Review the readme documentation provided with FixPak 4 for details.

- A local logon is not required to access the remote databases if the user ID and password values are used in the EZECONCT statement parameters.
This is true because we were using a client workstation that had only DB2 Client Application Enabler (CAE) installed.
If we did not pass the user ID and password values in the EZECONCT statement parameters, we were prompted for node logon.
- We found no changes in the previously reviewed functions for the database environment variable and profile settings.

The third configuration used remote DB2/2 and DB2/MVS databases where each was cataloged on the same node (see Figure 105). We used the client platform where we had installed DB2/2 CAE support.

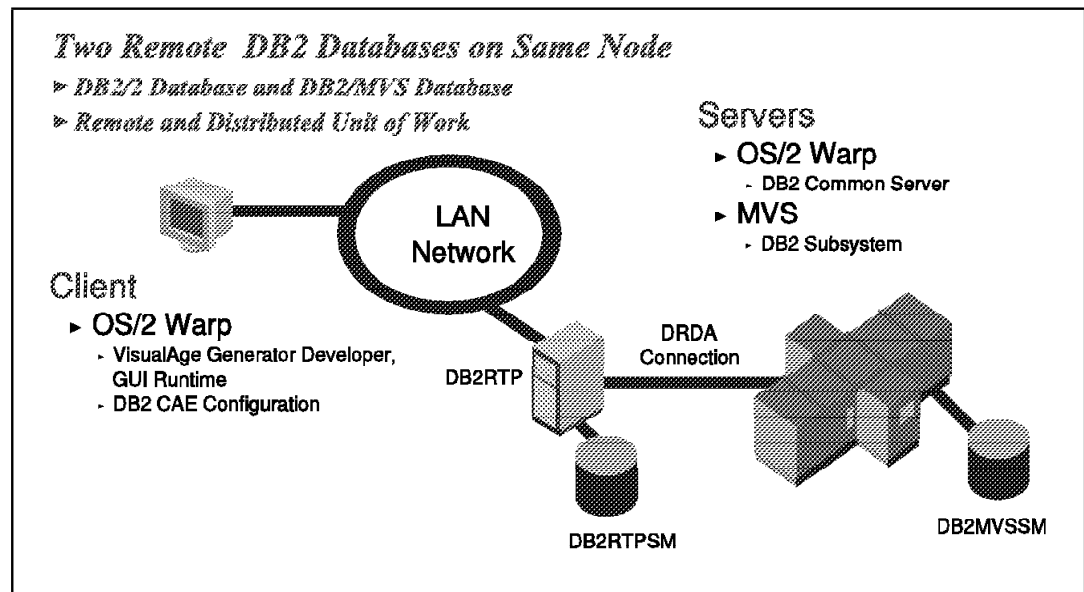


Figure 105. DUOW Configuration with Remote DB2/2 and Remote DB2/MVS Databases Cataloged on the Same Node

The DUOW sample application can now be tested in an ITF or native OS/2 runtime environment. We ran subsets of the standard test set (Figure 99 on page 127) using both of these runtime environments. Our tests focused on local and node logon requirements.

We learned the following:

- A local logon is not required to access the remote databases if the user ID and password values are used in the EZECONCT statement parameters. This is because we were using a client workstation that had only DB2 client application enabler (CAE) installed.
If we did not pass the user ID and password values in the EZECONCT statement parameters, we were prompted for node logon.
- Confusion resulted when we ran the DUOW sample application against two databases, cataloged on the same node, that did not share a common database authorization ID. The user ID and password used for each database was different; a single node logon was not valid for both targets. Although UPM did allow us to log on to the same node twice, with different user ID and password values, the DUOW sample application could not successfully access both databases without hard coding the required user ID and password data in the EZECONCT statement. If we did not pass the user ID and password in the EZECONCT statement, we would be prompted for multiple logons to the same node. It seemed as though the last node logon was current, and each test cycle forced a logon to the node

with two different authorization IDs, so we had to log on again to make the new logon current.

- Our distributed database configuration did not support distributed unit-of-work activity against the DB2/MVS database.

We could successfully use the EZECONCT statement to issue Type D2A connections to both target databases and perform an SQL SELECT. When we tried to make an update to both the DB2/2 and DB2/MVS target databases, the request failed with this error message:

SQL30090N

Operation invalid for application execution environment. Reason code = "<reason-code>".

We had reason code 01 which, according to the message manual, said our target database (the DB2/MVS database) was read only. Our best guess is that our DDCS/2 configuration, built on top of a CM/2 to VTAM link, was not able to support a two-phase commit. We were not able to resolve this during the residency project. A distributed database connection to DB2/MVS that does support a two-phase commit would support our DUOW sample application.

Note: DUOW processing requests, both SQL SELECT and UPDATE, succeeded against all other remote database pairs that we tested. As far as we could tell, the problem (SQL30090N) was with our database configuration and not the VisualAge Generator DUOW sample application design or preparation process.

5.2.3.9 Common Problems

Some of the problems encountered during our database configuration and DUOW sample application testing are discussed in Table 7. Many of the problems and resolution guidance represent pure database issues, but they often surface when using VisualAge Generator ITF for testing or running generated applications. Some problems we did not fully understand. They often disappeared when we reset the testing environment and we could not recreate them.

Table 7 (Page 1 of 2). Common Problems Encountered During DUOW Testing	
Error Message	Situation and Resolution
SQL0551N: <authorization-ID> does not have the privilege to perform operation <operation> on object <name>.	The user ID and password we used did not have the required authority. Grant authority to access the object or execute the package.
SQL0567N: <authorization-ID> is not a valid authorization ID.	We were testing the use of a single node for two remote databases. During our test cycles, we tried too many variations with the use of node logon or passed parameters for user ID and password resolution. If we were consistent in our approach and passed values for distinct nodes and node logons, this error did not occur.
SQL0805N: Package "<package-name>" was not found.	Our DUOW sample application had never been bound against the target database. All we had to do was bind the application.
SQL0818N: A timestamp conflict occurred.	Our DUOW sample application had been regenerated but not bound against the target database again. All we had to do was bind the application.

Table 7 (Page 2 of 2). Common Problems Encountered During DUOW Testing	
Error Message	Situation and Resolution
SQL0859N: Access to the Transaction Manager Database failed with SQLCODE "-865". SQLSTATE=08502.	We were testing the remote access option of the DUOW sample application with a target database of DB2MVSSM (uses a DRDA configuration). We set the TM_Database set to First_used. When we looked up "-865" the text stated that the TM_Database could not be accessed through a DRDA protocol. This also occurred when we had not configured the DB2 client with a transaction manager database setting. There is no initial value after installation; one must be configured before you can issue DUOW requests.
SQL0428N: DISCONNECT cannot be issued if a connection it is directed against has executed SQL within the unit of work. SQL0868N: A CONNECT with a USER/USING clause was attempted for a server for which a connection already exists. SQL0900N: The application state is in error. A database connection does not exist. SQL1024N: A database connection does not exist.	Many of these errors occurred when we were testing early versions of the DUOW sample application that had database-connection logic problems. Some were triggered when we attempted to process SQL SELECT statement even though our database connection request failed. Others were related to a lack of a firm commit or rollback request at application termination. We also triggered a few when we tried too many variations with the use of node logon or passed parameters for user ID and password resolution. Updates to the DUOW sample application seem to have removed these errors.
SQL1013N: The database alias name or database name "<name>" could not be found.	We used a database name that was not cataloged correctly or the name of the database in the program was spelled wrong.
SQL1246N: Connection settings cannot be changed while connections exist.	This happened for two reasons: First, when we were testing early versions of the DUOW sample application, we had database connection logic problems. Second, any time we issued a Type D2A connection request after a previous Type R connection request, without issuing a database disconnect, the error message appeared.
SQL1403N: The supplied user name, password, or both is incorrect.	This error occurred several times with the AIX and OS/2 target databases, but was because we provided a bad user ID or password value.
SQL30082N: Attempt to establish connection failed with security reason "<reason-code>" ("<reason-string>")	This error occurred several times with the MVS target database. Sometimes it was because we provided a bad user ID or password value. Once, it resulted from an expired password and was resolved by logging on to TSO to change the password.
SQL30090N: Operation invalid for application execution environment. Reason code = "<reason-code>".	We received reason code 01, which the message manual said meant that our target database, the DB2/MVS database, was read only. Our best guess is that our DDCS/2 configuration, built on top of a CM/2 to VTAM link, was not able to support a two-phase commit. We were not able to resolve this during the time-frame of our residency project.

Chapter 6. DBCS-Enabled Client/Server Configurations

This chapter reviews VisualAge Generator support for double-byte character set (DBCS) and mixed data applications. The implementation of a client/server application with DBCS support for data display and relational database access is documented for two different CICS-based configurations.

DBCS application design and development issues are discussed in detail in *Designing and Developing VisualAge Generator Applications*, SH23-0228.

6.1 DBCS Overview

DBCS requirements are prevalent in Asia where the number of characters in the language, along with other required symbols and numbers, exceeds the limits represented by a single byte of storage, thus changing the data storage and display processing requirements in a computing environment.

When an operating system provides support for a DBCS code page environment, the tools, products, or applications running on the operating system must be at least DBCS-enabled in order to support DBCS data storage and display.

There are three areas where DBCS implementation issues are visible: code page, data storage in files or databases, and I/O devices. Code pages issues are different for each supported language DBCS. Some computing platforms have multiple code pages for the same language.

In some language environments, there are two choices for DBCS support at the operating system level:

- Full DBCS-enabled operating system (Such as OS/2 J or OS/2 K)
Full DBCS-enablement means the kernel of the operating system has implemented full DBCS support.
- DBCS shell for single-byte character set (SBCS) operating system
A DBCS shell provides support for running an application which handles DBCS language support on a SBCS operating system platform.

IBM provides operating systems and other products that are designed to support application development and operation in DBCS code page environments.

VisualAge Generator is a DBCS-enabled product in that you can build applications that support DBCS code pages. VisualAge Generator is also translated into several DBCS languages:

- Japanese
- Chinese
- Korean

6.1.1 Starting BookManager in a DBCS Environment

Even though IBM supplies national language support (NLS) versions of VisualAge Generator, which includes translation of product dialogs, helps, and documentation, some programmers prefer to install the English language version of VisualAge Generator for DBCS application development. This choice works fine for most application development tasks.

The only exception is the tool used to provide online access to product manuals. The READIBM.EXE program, which is based on the BookManager product and delivered as an installable program with VisualAge Generator, needs special treatment when used in a DBCS-enabled operating system.

To use the English language version of the READIBM.EXE program on a DBCS-enabled operating system, the active code page must be changed so that it is compatible with the executable program. Code pages of 437 or 850 provide support for an English-language version of the READIBM.EXE program. The active code page for an OS/2 session can be different from the code page used by other active programs.

To identify the active code page, you can enter the `chcp` command at an OS/2 command prompt.

To change the active code page and use the READIBM.EXE to view online manuals, enter these commands at an OS/2 command prompt:

```
chcp 437
readibm
```

The READIBM.EXE will display a list of books and book shelves that are found using the current setting of the READIBM environment variable.

6.1.2 DBCS Data Definition

During design and development of a DBCS-enabled application, you must consider how SBCS and DBCS data types will be used. For VisualAge generator, some special considerations apply when defining data items, tables, maps, or applications that will support DBCS character entry and manipulation.

VisualAge Generator data-item definition provides two data types that support DBCS application requirements:

DBCS Data item can contain only double-byte characters; each single character is represented by 2 bytes of storage.

Mix Data item can contain both DBCS and SBCS data.

When a data item can contain both SBCS and DBCS data, special considerations apply. In an EBCDIC environment, special shift-out (SO) and shift-in (SI) characters are used to identify the start and end of DBCS strings in a mixed data-type field. The SO and SI characters add to the length of the data stored in a field. SO and SI characters are not required in an ASCII environment.

The length of data items should be defined to allow for the inclusion of these special characters in an EBCDIC environment. In a client/server system, if a MIX or DBCS data type field is used in both an ASCII and EBCDIC runtime environment, the addition of the SO and SI characters may make the string in the field longer in EBCDIC. You must consider the length of mixed data fields and data items during design and implementation so that space is reserved for the SO and SI characters.

For example, if you decide that the maximum number of characters in the DBCS data-type field 'NAME' will be 10, then the DB2 column in EBCDIC should be defined as GRAPHIC(12) to allow for SO and SI characters. If the data item is to be defined

with the mixed data type, the length calculation is a bit more complicated. Every time you change from a SBCS character to a DBCS character (or the reverse) in the same field, SO and SI characters are automatically added around each separate DBCS string. The length of the mixed data item should reflect the possible number of separate DBCS strings in the mixed field.

DBCS-enabled device types must be defined for map definitions in applications that provide 3270-type displays or support printing of DBCS data. For example:

Screen map	5550D 24*80 DBCS Display - TUI definition
Printer map	5550D 24*80 DBCS Printer - Printer definition

6.1.3 Relational Database Considerations

DBCS requirements also impact relational database definitions. VisualAge Generator DBCS data items correspond to either GRAPHIC or VARGRAPHIC column definitions in a DB2 environment while a data item defined with a mixed data type uses a CHAR column definition.

When importing a table originally created on a system with a different code page (and then exported), you may have to use the forcein import parameter to bypass code page conversion.

Try changing the code page (chcp command) before using database import and export commands or use the appropriate DB2 utility options to manage database import and export processing.

6.2 Client/Server Implementation for DBCS Applications

This section describes the VisualAge Generator DBCS-enabled client/server configurations that we implemented during the residency project.

In a VisualAge Generator client/server environment, DBCS application concerns include those that exist for SBCS applications and those concerns related to the use of different code pages, the impact on client/server environment setup, data type definition, and device selection for terminal and printer map input/output.

Unless specified, the steps discussed for setup are also suitable for SBCS code page environments.

6.2.1 DBCS Client/Server Scenarios

We implemented two DBCS-enabled client/server configurations:

Two-tier (client/server)

The client workstation uses CICS Client software and TCP/IP to connect to a server workstation running CICS OS/2. The server call is implemented on the CICS OS/2 platform. A CICS OS/2 VisualAge Generator server application performs database access as requested by the client. The database is a local DB2/2 database on the server workstation.

Three-tier (client/gateway/server)

The client workstation uses CICS client software and TCP/IP to connect to a server workstation running CICS OS/2. The server call is routed through CICS OS/2 (acting as a host gateway) to MVS CICS. An MVS CICS VisualAge Generator

server application performs database access as requested by the client. The CICS OS/2 and the MVS CICS host platform use an SNA LU6.2 connection implemented with CM/2.

Figure 106 shows these configurations.

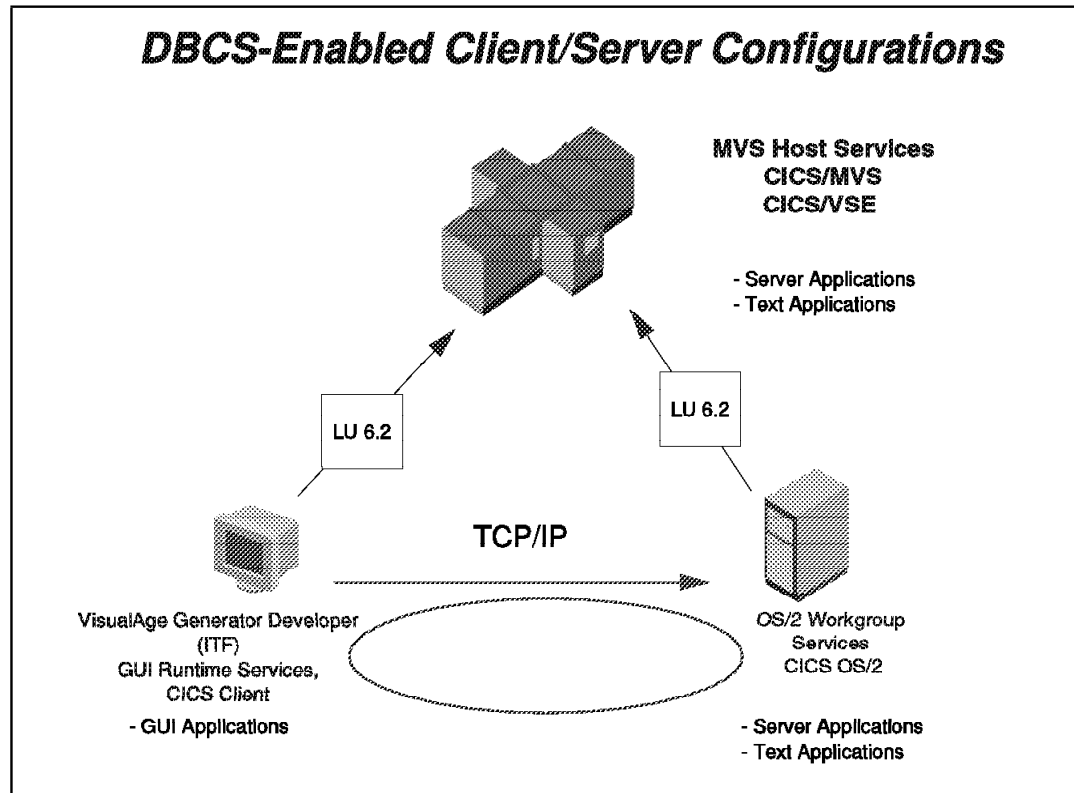


Figure 106. DBCS Client/Server Configuration Overview

6.2.1.1 Hardware and Software

We used the following hardware platforms and software configurations to implement our DBCS-enabled client/server configuration:

Client Workstation	
Hardware:	
Type	PC 750-P100
Hard drive	2.0 GB
RA MB	48 MB
Network Adapter	16/4 Token Ring Card
Software:	
Operating System	P-WARP (Simplified Chinese Version)
Enabling Software	P-CICS Client V1.1 VisualAge Generator GUI Run time for OS/2 Version 2.2 DB2/2 Client Application Enabler V2.1.1
IP address	9.37.196.59
Hostname	schinese

Server Workstation

Hardware:

Type	PS/2 95
Hard drive	1.0 GB
RAM	32 MB
NetWork Adapter	16/4 MCA Token Ring Card

Software:

Operating System	P-WARP (Simplified Chinese Version)
Enabling Software	P-CICS OS/2 Server V3.0 Communications Manager/2 V1.1.1 VisualAge Generator Developer V2.2 VisualAge Generator Workgroup Services V2.2 DB2/2 Server V2.1.1 DB2/2 SDK V2.1.1 VisualAge COBOL V1.1 (with available Fixpaks)
IP address	9.37.200.70
Hostname	china1

Host Platform

Hardware:

Not applicable

Software:

Operating System	MVS
Enabling Software	MVS CICS V3.3 DB2/MVS V2.3 VisualAge Generator Host Services V1.1 (CHS version)

The host session was configured to support DBCS application preparation using the command `cspcmds nls(P)`. The `cspcmds` command is a customized CLIST to allocate the correct set of APVFILES required for DBCS file transfer.

6.2.1.2 DBCS Sample Application Specifications

A small DBCS-enabled sample application was implemented. The application provided record maintenance for a DB2 table using the DBCS-enabled GUI window shown in Figure 107 on page 144



Figure 107. DBCS-Enabled Sample Application GUI Window

The DBCS sample application GUI called a server running on either CICS OS/2 or MVS CICS. This server updated the DB2 sample table STAFF.

The definition for the standard DB2 sample table (STAFF) was modified to provide support for DBCS data in two different columns. The NAME column was defined with support for variable-length graphic data (pure DBCS characters). The JOB column was defined so that it could contain both DBCS and SBCS character data. Figure 108 contains the definition of the DBCS-enabled staff table.

TBCREATOR	TBNAME	NAME	COLTYPE	LENGTH	SCALE	COLNO
JOAN	STAFF	ID	SMALLINT	2	0	1
JOAN	STAFF	NAME	VARG	10	0	2
JOAN	STAFF	DEPT	SMALLINT	2	0	3
JOAN	STAFF	JOB	CHAR	5	0	4
JOAN	STAFF	YEARS	SMALLINT	2	0	5
JOAN	STAFF	SALARY	DECIMAL	7	2	6
JOAN	STAFF	COMM	DECIMAL	7	2	7

Figure 108. STAFF Table Definition for DBCS-Enabled Environment

We used the create statement shown in Figure 109 on page 145 to create the STAFF table on the DB2 MVS system.

```

CREATE TABLE STAFF ( ID SMALLINT NOT NULL, NAME VARGRAPHIC(10),
                     DEPT SMALLINT, JOB CHAR(5) FOR MIXED DATA,
                     YEARS SMALLINT, SALARY DEC(7,2), COMM
                     DEC(7,2) );
INSERT INTO JOAN.STAFF VALUES ( 10, '邓淑珍', 80, 'TSS', 9, 88880.00, 800 );
GRANT ALL ON TABLE TO PUBLIC ;

```

Figure 109. Create Statement for DBCS-Enabled STAFF Table

The VisualAge Generator record definition for the STAFF table used a data type of DBCS for the NAME data item and MIXED for the JOB data item.

6.2.2 Environment Setup and Customization

To implement the scenarios discussed in 6.2.1, “DBCS Client/Server Scenarios” on page 141, we need to customize the software used to define and build the DBCS sample application and implement the network connections between the client and server workstations and then the server workstation and the host CICS platform.

The server workstation is used to support VisualAge Generator application development and generation, compilation of COBOL programs for the CICS OS/2 platform, and runtime support for CICS OS/2 applications. The server workstation is also configured as a CICS OS/2 gateway to MVS CICS applications.

For a typical application development effort, environment you should consider using separate workstations for development, generation, and CICS OS/2. Many of the steps we performed are similar to those described in 3.2.1, “CICS OS/2 Server” on page 29 and 3.2.3, “CICS Client” on page 44. Unless specified, the following customization steps are same as for other client/server configurations.

6.2.2.1 VisualAge COBOL

Only the VisualAge COBOL compiler needs to be installed. We generated the COBOL with VisualAge Generator; we use VisualAge COBOL just to prepare the program for execution.

We made the following change to the OS/2 CONFIG.SYS file on the server workstation to configure our target COBOL language system:

```
SET LANG=ZHCN1381
```

The default setting was ENUS437. A setting of ZHCN1381 provides support for Simplified Chinese.

6.2.2.2 VisualAge Generator on Client Workstation

We made the following changes to the OS/2 CONFIG.SYS file on the client workstation to configure our VisualAge Generator client/server runtime system control files and options:

```

SET CSOLINKTBL=C:\WORKTMP\WORKDIR\LNKTBL.TBL
SET CSOUEXIT=CSOPRMP
SET CSOTROPT=2
SET CSOTROUT=C:\WORKTMP\TRACE.OUT
SET CSOTIMEOUT=60

```

The CSOLINKTBL setting identifies the active linkage table. The linkage table on the client was used to support GUI calls to server applications. On the server workstation, the linkage table was used during generation processing.

The CSPTROPT and CSOTROUT settings identify the client/server communication trace options and output location for the trace file created during VisualAge Generator client/server call processing.

The CSOUEXIT setting identifies the exit we use to manage user ID and password identification and authentication. The CSOPRMPT exit uses a GUI window to prompt for the user ID and password values. These are passed to CICS as part of the server call through the CICS Client ECI interface.

The EZERSQLDB setting identifies the default database name that will be used when SQL requests are processed using VisualAge Generator Developer or generated VisualAge Generator applications.

6.2.2.3 VisualAge Generator on Server Workstation

We made the following changes to the OS/2 CONFIG.SYS file on the server workstation to configure our CICS and VisualAge Generator client/server runtime environment:

```
SET CICSWRK=C:\WORKTMP
SET EZERSQLDB=SAMPLE
```

The CICSWRK setting identifies the directory CICS OS/2 will use to find program executables.

VisualAge Generator applications are generated on the server workstation. The same linkage table must be accessible by (or replicated on) both the client and server workstations. The client workstation uses the linkage table to determine what kind of call to make and whether data conversion is required. The server workstation uses the linkage table during VisualAge Generator application generation.

VisualAge Generator Workgroup Services must be configured to support application preparation and execution for a CICS OS/2 target platform. To do this, we change the VisualAge Generator Workgroup Services file C:\VGWGS2\EXE\ELAENV.CMD as shown:

```
os2_install_drive = 'C:'
ela_install_drive = 'C:'
cics_install_drive = 'C:'
cobol_install_drive='C:'
ela_install_dir = '\VGWGS2'
cics_install_dir = '\CICS300'
cobol_install_dir = '\IBMCOBOL'
default_ezernls='CHS'
default_database = ''
ela_bit_mode='32'
ela_cics_group='VGWGS'
cics_version='3.0'
call_cicsenv=1
user_work='C:\worktmp'
cobol_type='IBM'
```

The modified ELAENV.CMD file identifies the drive and directory where VisualAge Generator Workgroup Services, CICS OS/2, and VisualAge COBOL are installed.

We used P-WARP as our operating system with an active code page of 1381 and support enabled for the 437 code page. The ELAENV.CMD file also identifies the NLS choice of Simplified Chinese (CHS).

See 3.2.1, "CICS OS/2 Server" on page 29 for additional information on configuring VisualAge Generator Workgroup Services for CICS OS/2.

If necessary, you can customize template files by entering:

```
C:\EZERDEV2\EFK2MBDB.TPL
```

Not all users have authority to create plans for accessing the database on the mainframe. You can change the DB-access plan directly into the template file. During configuration, we use the existing plan for the mainframe. Also, check what kinds of TRANSID the plan supports. You can use only one of those TRANSIDs to communicate with the remote system.

As for option files, the option file for SAMPGUI is C:\EZERDEV2\IEFKOP2GUI.OPT (default file of VG for CICS/2 for OS/2). The option file for DB2SAMP when it runs on MVS CICS is C:\EZERDEV2\IEFKOPMCS.OPT (default file of VG for MVS CICS), but change the IEFKOPMCS.OPT file as shown in Figure 110.

```
/* **** */
/*          Sample Options file for MVS CICS          */
/* **** */

/CICSENTRIES=NONE
/DATE=31
/JOB CARD=IEFK2MJOB.TPL
/NOICSDBCS
/NOENDCOMMAREA
/NOGENRET
/PRINTDEST=EZEPRINT
/SYNCDXFR
/SYSTEM=MVSCICS
/TWAOFF=0
/WORKDB=AUX
/SESSION=C
/PROJECTID=JOAN
/SYMPARM=ELA,' VGEN.HS.V1R1M0'
/SYMPARM=DSYS,' DSNA'
```

Figure 110. VisualAge Generator MVS CICS Generation Options

The JOBCARD is C:\EZERDEV2\TEMPLATES\IEFK2MJOB.TPL (default file of VG for MVS CICS) but change the IEFK2MJOB.TPL to read:

```
//JOANY JOB (997512,,,,N),CLASS=A,MSGCLASS=T,NOTIFY=&SYSUID
//PROC JCLLIB ORDER=(VGEN.HS.V1R1M0.PROCLIB)
```

We opened four sessions in CM/2 and opened a C session to transfer files. It is important that the TWAOFF size be smaller than TWA size in CICS OS/2.

Import the Workgroup Services CICS groups into CICS OS/2 as follows:

- At the OS/2 command prompt;

```
copy C:\VGWGS\ELAEX300.BTR C:\CICS300\RUNTIME\DATA\FAAAEFIE.BTR
```

- Start up CICS from Workgroup Services folder
- Log on with user SYSAD and password SYSAD (CICS OS/2 default user ID and password).
- Enter the CAIM transaction ID and press enter to display import panel.
- Enter the following information:

Field	Value
Group Name	VGWGS
Include Conversion template	N
Include SNT	N
Include data files	Y
Input from Backup file	N
Backup Existing Data	N

- Press Enter to start import.
- Shut down CICS ('cquit' transaction).
- Restart CICS OS/2 to activate changes
- Use the VisualAge Generator Workgroup Services ELAM transaction to verify the import operation and make sure that VisualAge Generator applications are ready to run.

6.2.2.4 Linkage Table Definitions

Our linkage table is defined in a file (C:\worktmp\workdir\linktbl.tbl) and is referenced during generation as well as by the CSOLINKTBL variable on the client workstation.

We actually needed different linkage table entries for each client/server configuration (two- and three-tier):

No conversion

When we called a CICS OS/2 server application, no ASCII/EBCDIC translation was performed.

Conversion

When an MVS CICS server was called, we needed to request ASCII/EBCDIC conversion in the linkage table entry and use the conversion table appropriate for Simplified Chinese.

The linkage table entry used for a two-tier client/server call is as follows:

```
:CALLLINK APPLNAME=* LINKTYPE=REMOTE PARMFORM=COMMDATA
      REMOTECOMTYPE=CICSCLIENT LUWCONTROL=SERVER LOCATION=CHINA1.
```

The linkage table entry used for a three-tier client/server call:

```
:CALLLINK APPLNAME=* LINKTYPE=REMOTE PARMFORM=COMMDATA
      REMOTECOMTYPE=CICSCLIENT LUWCONTROL=SERVER LOCATION=CHINA1
      CONTABLE=ELACNCHS.
```

ELACNCHS is the name of the Simplified Chinese conversion table provided by VisualAge Generator.

To provide complete support for the 1381 code page, we changed the 44th byte in the C:\EZERDEV2\ELACNCHS.CTB conversion table to a 1 as follows:

```
Original file: ELAASCTBELACNCHS05/01/95A00008361115083713800000101000000000000000000
Changes:                                     1381
```

6.2.2.5 Communications Manager/2

CM/2 should be configured to provide the APPC support required to connect CICS OS/2 on the server workstation with MVS CICS on the host.

The Communications Manager Setup program (CMSETUP.EXE) provides the dialog panels required to define an APPC session with the MVS CICS target. We defined a CM/2 configuration named MVSCICS. The CM/2 configuration can be viewed in the ASCII file C:\CMLIB\MVSCICS.NDF.

The following definitions are required in CM/2 to implement an APPC (LU6.2) configuration:

- One DLC Adapter Parameter definition to specify data link control characteristics of the network adapter.
- One local node characteristics definition to specify characteristics that are common to all APPC connections on the workstation.
- At least one connection definition.
- A local logical unit definition for each LU alias specified in a TCS entry that defines a remote CICS system to CICS OS/2.
- A partner logical unit definition for each remote CICS system.
- One or more MODE definitions to specify sets of session properties that are used in binding APPC sessions.
- A transaction program definition for every local transaction that can be invoked in an inbound request from a remote system.

Figure 111 on page 150 contains the MVSCICS.NDF file with APPC session support for the server workstation connection to MVS CICS.

```

DEFINE_LOCAL_CP FQ_CP_NAME(USIBMNR.NRRMKJ00 )
DESCRIPTION(Local node)
CP_ALIAS(NRRMKJ00)
NAU_ADDRESS(INDEPENDENT_LU)
NODE_TYPE(EN)
NODE_ID(X'05D00000')
NW_FP_SUPPORT(NONE)
HOST_FP_SUPPORT(YES)
HOST_FP_LINK_NAME(HOST0001)
MAX_COMP_LEVEL(NONE)
MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK LINK_NAME(HOST0001)
DESCRIPTION(Connection to the USIBMNR Network)
FQ_ADJACENT_CP_NAME(USIBMNR.NRMC1 )
ADJACENT_NODE_TYPE(LEN)
DLC_NAME(IBMTRNET)
ADAPTER_NUMBER(0)
DESTINATION_ADDRESS(X'400020600632')
ETHERNET_FORMAT(NO)
CP_SESSION_SUPPORT(NO)
SOLICIT_SSCP_SESSION(YES)
NODE_ID(X'05D00000')
ACTIVATE_AT_STARTUP(YES)
USE_PUNAME_AS_CPNAME(NO)
LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
COST_PER_BYTE(USE_ADAPTER_DEFINITION)
SECURITY(USE_ADAPTER_DEFINITION)
PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
USER_DEFINED_1(USE_ADAPTER_DEFINITION)
USER_DEFINED_2(USE_ADAPTER_DEFINITION)
USER_DEFINED_3(USE_ADAPTER_DEFINITION);
DEFINE_LOCAL_LU LU_NAME(NRMKJ00I)
DESCRIPTION(Local LU)
LU_ALIAS(NRMKJ00I)
NAU_ADDRESS(INDEPENDENT_LU);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMNR.NRAC1S1 )
DESCRIPTION(NRAC1S1 CICS on CARMVS1)
PARTNER_LU_ALIAS(NRAC1S1)
PARTNER_LU_UNINTERPRETED_NAME(NRAC1S1)
MAX_MC_LL_SEND_SIZE(32767)
CONV_SECURITY_VERIFICATION(NO)
PARALLEL_SESSION_SUPPORT(YES);

```

Figure 111 (Part 1 of 3). CM/2 NDF File for Server Workstation Host Connection

```

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(USIBMNR.NRARDSNA )
PARTNER_LU_ALIAS(NRARDSNA)
PARTNER_LU_UNINTERPRETED_NAME(NRARDSNA)
MAX_MC_LL_SEND_SIZE(32767)
CONV_SECURITY_VERIFICATION(NO)
PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(USIBMNR.NRACICS1 )
DESCRIPTION(NRACICS1 CICS on CARMVS1)
WILDCARD_ENTRY(NO)
FQ_OWNING_CP_NAME(USIBMNR.NRMCMC1 )
LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(USIBMNR.NRARDSNA )
WILDCARD_ENTRY(NO)
FQ_OWNING_CP_NAME(USIBMNR.NRMCMC1 )
LOCAL_NODE_NN_SERVER(NO);

DEFINE_MODE  MODE_NAME(LU62APPX)
DESCRIPTION(LU62 Logmode)
COS_NAME(CPSVCMG )
DEFAULT_RU_SIZE(NO)
MAX_RU_SIZE_UPPER_BOUND(1024)
RECEIVE_PACING_WINDOW(4)
MAX_NEGOTIABLE_SESSION_LIMIT(32767)
PLU_MODE_SESSION_LIMIT(8)
MIN_CONWINNERS_SOURCE(2)
COMPRESSION_NEED(PROHIBITED)
PLU_SLU_COMPRESSION(NONE)
SLU_PLU_COMPRESSION(NONE);
DEFINE_MODE  MODE_NAME(IBMDB2LM)
DESCRIPTION(LU6.2 Log Mode for DB2)
COS_NAME(#CONNECT)
DEFAULT_RU_SIZE(NO)
MAX_RU_SIZE_UPPER_BOUND(4096)
RECEIVE_PACING_WINDOW(8)
MAX_NEGOTIABLE_SESSION_LIMIT(32767)
PLU_MODE_SESSION_LIMIT(4)
MIN_CONWINNERS_SOURCE(2)
COMPRESSION_NEED(PROHIBITED)
PLU_SLU_COMPRESSION(NONE)
SLU_PLU_COMPRESSION(NONE);
DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
DEFAULT_MODE_NAME(BLANK)
MAX_MC_LL_SEND_SIZE(32767)
DIRECTORY_FOR_INBOUND_ATTACHES(*)
DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
DEFAULT_TP_CONV_SECURITY_RQD(NO)
MAX_HELD_ALERTS(10);

```

Figure 111 (Part 2 of 3). CM/2 NDF File for Server Workstation Host Connection

```

DEFINE_TP TP_NAME(CPMI)
PIP_ALLOWED(NO)
FILESPEC(C:\CICS300\RUNTIME\FAACLPIN.EXE)
PARM_STRING(CPMI)
CONVERSATION_TYPE(ANY_TYPE)
CONV_SECURITY_RQD(NO)
SYNC_LEVEL(EITHER)
TP_OPERATION(NONQUEUED_AM_STARTED)
PROGRAM_TYPE(BACKGROUND)
RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

DEFINE_TP TP_NAME(CVMI)
PIP_ALLOWED(NO)
FILESPEC(C:\CICS300\RUNTIME\FAACLPIN.EXE)
PARM_STRING(CVMI)
CONVERSATION_TYPE(ANY_TYPE)
CONV_SECURITY_RQD(NO)
SYNC_LEVEL(EITHER)
TP_OPERATION(NONQUEUED_AM_STARTED)
PROGRAM_TYPE(BACKGROUND)
RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

DEFINE_TP TP_NAME(CRSR)
PIP_ALLOWED(NO)
FILESPEC(C:\CICS300\RUNTIME\FAACLPIN.EXE)
PARM_STRING(CRSR)
CONVERSATION_TYPE(ANY_TYPE)
CONV_SECURITY_RQD(NO)
SYNC_LEVEL(EITHER)
TP_OPERATION(NONQUEUED_AM_STARTED)
PROGRAM_TYPE(BACKGROUND)
RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

DEFINE_TP TP_NAME(CRTE)
PIP_ALLOWED(NO)
FILESPEC(C:\CICS300\RUNTIME\FAACLPIN.EXE)
PARM_STRING(CRTE)
CONVERSATION_TYPE(ANY_TYPE)
CONV_SECURITY_RQD(NO)
SYNC_LEVEL(EITHER)
TP_OPERATION(NONQUEUED_AM_STARTED)
PROGRAM_TYPE(BACKGROUND)
RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

DEFINE_CPIC_SIDE_INFO SYMBOLIC_DESTINATION_NAME(NRARDSNA)
    DESCRIPTION(CPIC Information for DB2 DSN subsystem)
    PARTNER_LU_ALIAS(NRARDSNA )
    MODE_NAME(IBMDB2LM)
    SNA_SERVICE_TP_NAME(X'07',6DB);

START_ATTACH_MANAGER;

```

Figure 111 (Part 3 of 3). CM/2 NDF File for Server Workstation Host Connection. The definitions included here provided support for a CICS OS/2 to MVS CICS connection (used to implement server calls) and a DB2/2 to DB2 MVS DRDA connection (remote database access). The remote database access connection was not used during the implementation of DBCS.

We have defined two APPC (LU6.2) sessions in our MVSCICS.NDF file. One is for MVS CICS, the other is for a DRDA connection from DB2/2 on the server workstation to DB2 MVS. The mode name for the MVS CICS connection is LU62APPX, while the mode name for the DB2 MVS connection is IBMDB2LM.

6.2.2.6 CICS Client

CICS Client is a DBCS-enabled product. You can choose to install either a DBCS-language version or the English language version.

A detailed review of the CICS Client configuration is available in 3.2.3, “CICS Client” on page 44.

We connected the CICS Client on our client workstation with CICS OS/2 on the server workstation using TCP/IP. Figure 112 on page 153 contains an extract of part of our CICS Client initialization file (CICSCLI.INI).

```

;*****
;* IBM CICS Client - Initialization File *
;*****
;
;-----
;Server section - This section defines a server to which the client may
;connect. There may be several Server sections.

; TCP/IP Connection to CICS OS/2
;
Server = CHINA1                ; Arbitrary name for the server
Description = TCP/IP Server    ; Arbitrary description for the server
Protocol = TCPIP               ; Matches with a Driver section below
NetName = 9.37.200.70          ; The server's TCP/IP address
Port = 0                       ; Use the default TCP/IP CICS port

; SNA (CM/2) Connection to MVS CICS
;
; Server = CICS1                ; Arbitrary name for the server
; Description = SNA Server      ; Arbitrary description for the server
; Protocol = SNA                ; Matches with a Driver section below
; NetName = USIBMNR.NRACICS1    ; The server's fully qualified LU name
; LocalLUName = NRMKJ00I        ; The client's local LU name
; ModeName = LU62APPX           ; The SNA communications mode name

;-----
; Driver section - This section defines a communications protocol DLL
; used to communicate with a server. There may be
; several Driver sections.
;
; The default example is for NetBIOS communications. Further examples
; for TCP/IP and SNA communications are shown but are commented out.

Driver = TCPIP                  ; Matches the Server's Protocol value
DriverName = CCLIBMIP           ; Use the IBM TCP/IP communications DLL

; Driver = SNA                  ; Matches the Server's Protocol value
; DriverName = CCLIBMSN         ; Use the IBM SNA communications DLL
;

```

Figure 112. CICS Client Initialization File

When we started the CICS Client program (CICSCLI), we used these commands:

```

CICSCLI /S=CHINA1 /F=CICSCLI.INI
CICSCLI /C=CHINA1 /U=sysad /P=sysad

```

The first command told the CICS Client that we wanted a session with the CHINA1 server (/S=CHINA1) and that we wanted to use a specific initialization file (/F=CICSCLI.INI). The second command told the CICS Client that we wanted to use a specific user ID and password (/U=sysad /P=sysad) for the CHINA1 connection (/C=CHINA1).

The user ID sysad, the default system administrator ID for CICS OS/2, was used during our testing.

The CICS Client initialization file shown in Figure 112 also contains support for a direct connection to MVS CICS. If we removed the comment markers (;) from the definitions for the CICS1 server and the SNA driver, we could connect directly to MVS CICS from the client workstation (if the appropriate CM/2 definitions have been made).

6.2.2.7 CICS OS/2 Server

Setup activity for the CICS OS/2 server will depend on which language version you have installed. See 3.2.1, "CICS OS/2 Server" on page 29 for a discussion on CICS OS/2 system setup and ways of managing CICS SIT definitions so that your SIT is independent of the default SIT provided by CICS OS/2.

If you have installed the English version of CICS OS/2, you need to change the TCT entry for the V123 terminal to define a code page value of 1381 (Simplified Chinese). If you have installed P-CICS OS/2, then 1381 is the default code page.

We used the CEDA transaction to customize CICS OS/2 to identify our system as a unique system that supports VisualAge Generator applications (SIT changes), to define a link to the remote MVS CICS system (TCS entry), and to provide support for calling a remote program on the MVS CICS system (PPT entry).

Figure 113 contains an extract of our modified CICS OS/2 SIT entry.

Group Name	FAASYS
Description	SAMPLE SYSTEM INITIALIZATION
System Sizes	
CWA size	0
Maximum TWA size	1024
Number of trace entries	200
Task Control	
Max number of tasks	12
Min # of free tasks	10
Task Classes	1 2 3 4 5 6 7 8 9 10
Maximum tasks in Class	1 1 1 1 1 1 1 1 1 1
Default Process Priority	86
CICS System Priority	0
System Communications	
Local System ID	CICI
Local System Appl id	CHINA1
Default Remote System ID	J00I
TCP/IP Support	
TCP/IP local host name	9.37.200.70
TCP/IP local host port	*
Maximum tasks in TCP/IP	10
Miscellaneous	
Load PNA Support	N
PNA Model Terminal	MPNA
Initial Transaction ID	CLOG
Dump on Abend	N
Date Format	MMDDYY
External File Manager Name	
User Conversion Table	

Figure 113. CICS OS/2 SIT Table Definitions

The key fields that we modified in the SIT are these:

Maximum TWA size

This must be set to at least 1024 to provide support for running VisualAge Generator Workgroup Services applications.

Maximum number of tasks

Equal to minimum free tasks plus the number of the CICS OS/2 video terminals. The default is two video terminals.

Minimum free tasks

Equal to the number of concurrent nonfacility tasks. These are tasks that do not use a permanent terminal as their principal facility. Client terminals are considered nonfacility tasks.

Local system ID

The local ID must match:

- The connection name in MVS CICS connection definition
- Session and connection name in MVS CICS session definition

Local System Application ID

Used in CICS Client definitions. The CICS OS/2 application ID is identified in the CICS Client initialization file (CICSCLI.INI) when defining a connection between the client and the target CICS OS/2 server.

Default remote system ID

If a CICS OS/2 server terminal or client issues a transaction that is not defined locally, CICS OS/2 will attempt to route the transaction to the CICS system specified by this parameter.

TCP/IP local host name

Identifies OS/2 server workstation TCP/IP name or address. If a name is provided, the TCP/IP name server must be configured.

CICS OS/2 requires a terminal control table system entry (TCS) for any remote CICS system with which it may communicate. Figure 114 contains the CICS OS/2 TCS entry for our remote MVS CICS system.

System ID	J001
Group Name	DBCSG
Connection Type	APPC
Connection Priority	086
Description	Communicate with MVS CICS NRARCICS1
Session Details	
Session Count	12
Session Buffer Size	1024
Attach Security	L
Partner Code Page	0037
APPC Details	
Mode name	LU62APPX
LU alias	NRMKJ001
Partner LU Alias	NRARCICS1

Figure 114. CICS OS/2 Terminal Control Table

The key fields in the terminal control system are:

System ID

The name of the remote system with which CICS OS/2 communicates.

Session count

The maximum number of concurrent sessions that can be active. This value should match the following parameters:

- Mode session limit in the CM/2 mode definition
- Maximum (first parameter) in the MVS CICS session definition

Session buffer size

The maximum buffer size for each session. The session buffer size should be the largest COMMAREA size we expect to send or receive in our sample application. This value should match the:

- Send size and Receive size in MVS CICS session definition
- Maximum request and response unit (RU) size in the CM/2 mode definition.
- RUSIZES in the VTAM MODEENT for LU62APPX

Attach security

L means local security, so that the user ID and password are not passed to the remote system. It assumes that the user is authorized.

Partner code page Page 37 is for U.S. EBCDIC used by our MVS CICS host.

Mode name

Defines the session properties for the LU6.2 sessions. It matches an entry in the CM/2 SNA features mode list.

LU alias

The alias of the local LU defined with CM/2. It is used by CICS OS/2 instead of the actual local LU name.

Partner LU alias

The name is used by CICS OS/2 instead of the fully qualified partner LU name to refer to the remote CICS system. CICS OS/2 uses this name to obtain the SNA network name of the remote system. All intersystem communications take place between the local LU specified by the LU alias, and the Partner LU specified by this Partner LU alias. This name must match the alias in the CM/2 partner LU definition.

Programs and map sets to be used in CICS OS/2 do not necessarily need an entry in the processing program table (PPT). If a program is called for which no PPT entry exists, CICS OS/2 will try to locate an executable version of the program. If an executable version of the program is found, it is loaded as if there had been a PPT entry.

Similarly, when a CICS INQUIRE PROGRAM STATUS command is issued for a program that is not described by a PPT entry, CICS OS/2 tries to locate it. If CICS OS/2 finds an executable version of the program it returns a status of ENABLED.

When we configure the DBCS sample application to call a VisualAge Generator server application in CICS, then OS/2 does not need a PPT entry. We could have defined a PPT, but no remote system is assigned.

To allow CICS OS/2 to issue a distributed program link to a program on a remote system, the program must be defined to CICS OS/2 as remote. When we configure the DBCS sample application to call an VisualAge Generator server application in MVS CICS, we need to define a PPT entry. Figure 115 on page 157 shows the CICS OS/2 PPT entry we use to call a remote program on our MVS CICS system.

Program Name	DB2SAMP
Group Name	DBCSG
Program(P)/Map(M)/Data(D)	P
Resident(P,T,N)/Remote(R)	R
Remote (R) SYSID	J00I
Remote (R) PROGRAM	
Remote (R) TRANSID	
Description	Application running on MVS CICS

Figure 115. Processing Program Table for CICS OS/2

The key fields are these:

Remote (R) SYSID Demands the name of a valid TCS entry defined to CICS OS/2.

Remote (R) TRANSID

If no value is provided, the call is implemented using the default transaction ID CPMI. To use an alternative transaction ID, you can provide a value that is defined on the remote system (MVS CICS in our configuration) with attributes that are similar to the CPMI transaction. This may be required if you are using separate resource control table (RCT) entries for different transactions.

Note: The RCT entry connects an MVS CICS transaction with DB2 resources. A plan name and other DB2 connection information is defined in the RCT entry.

We used the default transaction CPMI, so we did not define any PCT entries in MVS CICS.

6.2.2.8 MVS CICS

MVS CICS customization can be performed using macros or the resource definition on-line (RDO) CICS function provided by the CEDA transaction. We used the CEDA transaction to to define the following resources to MVS CICS:

- Connection
- Session
- Transaction
- Program

Note: The authority to create and modify MVS CICS resource definitions may be restricted, so you may have to ask the CICS system administrator for assistance.

The connection and session resources are used to communicate with CICS OS/2. The transaction and program resources are used to support the remote call of a VisualAge Generator server application from CICS OS/2.

Figure 116 on page 158 contains the MVS CICS connection definition for the remote CICS OS/2 system.

```

OBJECT CHARACTERISTICS
CEDA View Connection( J00I )
Connection      : J00I
Group           : DBCSG
Description      : SIMPLIFIED CHINESE(LAB)
CONNECTION IDENTIFIERS
Netname         : NRMKJ00I
INDsys          :
REMOTE ATTRIBUTES
REMOTESYSTEM     :
REMOTENAME       :
REMOTESYSNET     :
CONNECTION PROPERTIES
Accessmethod     : Vtam          Vtam | IRc | INdirect | Xm
Protocol         : Appc          Appc | Lu61 | Exci
Connntype        :              Generic | Specific
Singlesess       : No           No | Yes
Datastream       : User          User | 3270 | SCs | STfield | Lms
RECORDformat     : U            U | Vb
QueueLimit       : No           No | 0-9999
Maxqtime         : No           No | 0-9999
OPERATIONAL PROPERTIES
Autoconnect      : No           No | Yes | All
INService        : Yes         Yes | No
SECURITY
Securityname     : HEATON
Attachsec        : Verify       Local | Identify | Verify | Persistent
                                   | Mixidpe
BINDPassword     :              PASSWORD NOT SPECIFIED
BINDSecurity     : No           No | Yes
Usedfltuser      : Yes         No | Yes
RECOVERY
PSrecovery       : Sysdefault   Sysdefault | None

SYSID=CICS APPLID=NRACICS1

```

Figure 116. MVS CICS Connection Definition for CICS OS/2 System

The key fields in the connection definition are these:

- | | |
|-------------------|---|
| Connection | <p>The entry must match the</p> <ul style="list-style-type: none"> Local System ID in the CICS OS/2 SIT Connection name in MVS CICS session definition |
| Netname | <p>The name of the independent LU. It must match the</p> <ul style="list-style-type: none"> LU name used in the CM/2 SNA features local LU definition. LU alias in the CICS OS/2 TCS entry for the remote CICS system <p>The NetName should be the CICS OS/2 LU Name.</p> |

Figure 117 on page 159 contains the MVS CICS session definition for the remote CICS OS/2 system.

OBJECT CHARACTERISTICS		CICS RELEASE = 0410	
CEDA View Sessions(J00I)			
Sessions	: J00I		
Group	: DBCSG		
DEscription	:		
SESSION IDENTIFIERS			
Connection	: J00I		
SESSName	:		
NETnameq	:		
MOdename	: LU62APPX		
SESSION PROPERTIES			
Protocol	: Appc	Appc Lu61 Exci	
MAximum	: 004 , 002	0-999	
RECEIVEPfx	:		
RECEIVECount	:	1-999	
SENDPfx	:		
SENDCount	:	1-999	
SENDSize	: 00256	1-30720	
RECEIVESize	: 00256	1-30720	
SESSPriority	: 000	0-255	
Transaction	:		
OPERATOR DEFAULTS			
OPERId	:		
OPERPriority	: 000	0-255	
OPERRsl	: 0		0-24,...
OPERSecurity	: 1		1-64,...
PRESET SECURITY			
USERId	:		
OPERATIONAL PROPERTIES			
Autoconnect	: Yes	No Yes All	
INservice	:		
Buildchain	: Yes	Yes No	
USERArealen	: 000	0-255	
IOarealen	: 00000 , 00000	0-32767	
RELreq	: No	No Yes	
DIScreq	: No	No Yes	
NEPclass	: 000	0-255	
RECOVERY			
RECOVOption	: Sysdefault	Sysdefault Clearconv Releasesess Uncondrel None	
RECOVNotify	: None	None Message Transaction	
SYSID=CICS APPLID=NRACICS1			

Figure 117. MVS CICS Session Definition for CICS OS/2 System

The key parameters are these:

Session	Ask for the name of the session.
Connection	Defines the name of the connection associated with this session. It must match the • Connection name in the MVS CICS connection definition for the remote CICS system • Local System ID in the CICS OS/2 SIT
Mode Name	The log mode entry name. It must match the MODEENT name in the VTAM MODETAB.
Maximum	The first value defines the maximum number of sessions supported. It must match the • Session count in the CICS OS/2 TCS entry for the remote CICS system • Mode session limit in the CM/2 SNA features mode definition The second value defines the maximum number of sessions which can be contention winners on the MVS CICS end of the connection.

The sum of this number and the maximum number of contention winners defined to CM/2 should equal the total for MAXIMUM sessions.

SEND SIZE	The RU size for sending data. This value should match the: • Maximum COMMAREA size that will be sent • Session buffer size in the CICS OS/2 TCS entry for the remote CICS system • Maximum RU size in the CM/2 SNA features mode definition • RUSIZES in the VTAM MODEENT definition
RECEIVE SIZE	The RU size for receiving data. This value should match all of these: • Maximum COMMAREA size to be received • Session buffer size in the CICS OS/2 TCS entry for the remote CICS system • Maximum RU size in the CM/2 SNA features mode definition • RUSIZES in the VTAM MODEENT definition.
AUTOCONNECT	YES allows CICS to establish a session automatically with the session partner during CICS system initialization

We used the default CICS mirror transaction CPML in our configuration. This definition needed to be modified to increase the TWA size as required by VisualAge Generator applications. Figure 118 on page 161 contains the modified MVS CICS transaction definition for the CPML mirror transaction.

OBJECT CHARACTERISTICS		CICS RELEASE = 0410
CEDA View TRANSAction(CPMI)		
TRANSAction	: CPMI	
Group	: DBCSG	
DEscription	: CICS OS/2 LU62 Mirror	
PROgram	: DFHMIRS	
TWAsize	: 01024	0-32767
PROfile	: DFHCICSA	
PARTitionset	:	
STatus	: Enabled	Enabled Disabled
PRIMedsize	: 00000	0-65520
TASKDATAloc	: Below	Below Any
TASKDATAkey	: User	User Cics
STOrageclear	: No	No Yes
RUnaway	: System	System 0-2700000
SHutdown	: Disabled	Disabled Enabled
ISolate	: Yes	Yes No
REMOTE ATTRIBUTES		
+ DYNamic	: No	No Yes
+ REMOTESystem	:	
REMOTENAME	:	
TRProf	:	
Localq	:	No Yes
SCHEDULING		
PRIOrity	: 001	0-255
TClass	: No	No 1-10
TRANClass	: DFHTCL00	
ALIASES		
Alias	:	
TASKReq	:	
XTRanid	:	
TPName	:	
	:	
XTPname	:	
	:	
+ RECOVERY		
DTImout	: No	No 1-6800
INDoubt	: Backout	Backout Commit Wait
REStart	: No	No Yes
SPurge	: No	No Yes
TPUrge	: No	No Yes
DUmp	: Yes	Yes No
TRACe	: Yes	Yes No
COnfdata	: Yes	No Yes
SECURITY		
RESec	: Yes	No Yes
CMdsec	: No	No Yes
Extsec	: No	
TRANSec	: 01	1-64
RS1	: 00	0-24 Public
SYSID=CICS APPLID=NRACICS1		

Figure 118. MVS CICS Mirror Transaction Definition

Figure 119 on page 162 contains the MVS CICS program definition for the VisualAge Generator server application that was defined as remote on the CICS OS/2 system (see Figure 115 on page 157).

OBJECT CHARACTERISTICS		CICS RELEASE = 0410
CEDA View PROGram(DB2SAMP)		
PROGram	: DB2SAMP	
Group	: DBCSG	
DEscription	:	
Language	: CObol	CObol Assembler Le370 C Pl i
		Rpg
RELoad	: No	No Yes
RESident	: No	No Yes
USAge	: Normal	Normal Transient
USEI pacopy	: No	No Yes
Status	: Enabled	Enabled Disabled
RSI	: 00	0-24 Public
Cedf	: Yes	Yes No
DAtalocation	: Below	Below Any
EXECKey	: User	User Cics
REMOTE ATTRIBUTES		
REMOTESystem	:	
REMOTEName	:	
Transid	:	
EXECutionset	: Fullapi	Fullapi Dplsubset
SYSID=CICS APPLID=NRACICS1		

Figure 119. MVS CICS Program Definition

Appendix A. Sample Applications

One of the goals of the project that produced this publication was to define and implement as many of the available client/server communication configurations supported by VisualAge Generator as possible during the residency period. Two basic types of client/server environments were studied:

- **Distributed function:** GUI client or local server calls a remote server. This is implemented in the VGPING sample application.
This approach required the use of multiple workstations and a choice between CICS-based or DCE-based client/server communications options.¹⁹
- **Remote or distributed database access:** Local application accesses one or more local or remote databases. This is implemented in the DUOW sample application.

Multiple database access included support for distributed unit of work management as implemented using the VisualAge Generator EZECONCT statement

A.1 VGPING Sample Application

To provide support for testing multiple distributed-function client/server scenarios, we designed and implemented the VGPING sample application.

A.1.1 Objectives

The objective of the VGPING sample application was to make our job of building and testing different client/server communication configurations easier. To do this, we built a custom VisualAge Generator client/server application that satisfies these functional objectives:

- One GUI application is used for all possible client/server communication configurations.
- Supports two- and three-tier server configurations.
- Provides default SQL support on the target platforms.
- Allows mixed ITF-based and generated execution of the sample applications.
- Captures VisualAge Generator provided return codes for all server application calls.
- Obtains user ID information using the VisualAge Generator EZEUSERID EZEword.
- Measures elapsed time for the two- or three-tier server call.

¹⁹ Although the IMS APPC, VisualAge Generator middleware, CA/400, and MQI approaches to client/server were discussed in this publication, we did not implement those scenarios.

A.1.2 Overview

The VGPIPING sample application developed provides support for testing client/server calls in both two- and three-tier client/server communication configurations. The VGPIPING client acts as the first tier and can be asked to call a server (second tier). The parameter data passed to this server can contain a request for database access and a request for a call to another server (third tier). This server can also be requested to perform database access.

The VGPIPING client can be run in any of these environments:

- ITF
- OS/2
- Windows 3.11 (Not implemented)
- Windows 95
- Windows NT

These client/server communication options and target environments are supported for calls to tier-2 servers from the client or tier-3 servers from tier-2 servers:

- ITF (no client/server communication required)
- CICS-based client/server communication with these target server platforms:
 - CICS OS/2
 - CICS NT
 - CICS/6000
 - MVS CICS*
 - VSE CICS (Not implemented)
- DCE-based client/server communication with these target server platforms:
 - OS/2
 - Windows NT
 - AIX
 - MVS CICS (Not implemented)
 - MVS IMS (Not implemented)
- IMS APPC, VisualAge Generator middleware, and CA 400 (Not implemented)
- * Discussed in Chapter 6, “DBCS-Enabled Client/Server Configurations” on page 139

For example, the VGPIPING sample application could support two-tier call scenarios such as:

- ITF GUI client calls a CICS/6000 server (CICS-based client/server communication).
- Windows 95 GUI client calls a Windows NT server (DCE-based client/server communication).
- OS/2 GUI client calls an AIX server (DCE-based client/server communication).

The VGPIPING sample application could also support three-tier call scenarios, such as:

- ITF GUI client calls a CICS OS/2 server which calls a CICS/6000 server (CICS-based client/server communication).
- Windows 95 GUI client calls a Windows NT server which calls an AIX server (DCE-based client/server communication).
- OS/2 GUI client calls an OS/2 server (DCE-based client/server communication) which calls a CICS/6000 server (CICS-based client/server communication).

Not all the potential call scenarios are possible or make sense in a production environment. This VGPIPING sample application capability made the task of implementing and testing multiple client/server configurations easier to perform. You can use the VGPIPING sample application as a simple but effective testing tool for your preferred client/server communication configurations.

The VGPING client does not edit your choice of servers to call. You can attempt to have a local OS/2 (tier-2) server call an ITF (tier-3) server, but such a request will fail. Calls attempted must be supported (CICS servers cannot call non-CICS servers) and the prerequisite client/server communication configuration and server generation steps must be performed.

Not all the possible scenarios have been implemented and tested. We believe that by including support for these configurations in the sample application we can help you define and test your desired client/server communication configuration.

A.1.3 Function

The VGPING sample application provides the following functions:

- Selection of a tier-2 server target (GUI calls local or remote server)
- Optional selection of a tier-3 server target (tier-2 server calls local or remote server)

The server name determines the position in the two- or three-tier call path and target runtime platform. The server names are shown in Table 8.

- Push button selection of the processing request:

Ping Call servers without any SQL processing. Tests only the client/server communication configuration.

Exit End the DUOW client session.

Table 8. Sample application Servers. The full server name and the name used in the VGPING client user interface are shown. The name in parentheses is the name used in the VGPING client selection lists. We defined, but did not implement, all the client/server configurations listed.

Client/Server Communication Option	Two-Tier Server (Name in sample application list box)	Three-Tier Server (Name in sample application list box)
ITF Only	VGL2ITF (L2ITF)	VGL3ITF (L3ITF)
Local Calls	VGL2OS2 (L2OS2)	VGL2WNT (L2WNT)
CICS	VGC2AIX (C2AIX)	VGC3AIX (C3AIX)
	VGC2MVS (C2MVS)	VGC3MVS (C3MVS)
	VGC2OS2 (C2OS2)	VGC3OS2 (C3OS2)
	VGC2VSE (C2VSE)	VGC3VSE (C3VSE)
	VGC2WNT (C2WNT)	VGC3WNT (C3WNT)
DCE	VDG2AIX (D2AIX)	VDG3AIX (D3AIX)
	VDG2IMS (D2IMS)	VDG3IMS (D3IMS)
	VDG2MVS (D2MVS)	VDG3MVS (D3MVS)
	VDG2OS2 (D2OS2)	VDG3OS2 (D3OS2)
	VDG2WNT (D2WNT)	VDG3WNT (D3WNT)
VisualAge Generator Middleware for CA/400	VGA2400 (A2400)	VGS3400 (S3400)
VisualAge Generator Middleware for IMS/VS	VGS2IMS (S2IMS)	VGS3IMS (S3IMS)
VisualAge Generator Middleware using TCP/IP	VGT2400 (T2400)	VGT3400 (T3400)
VisualAge Generator Middleware using Named Pipes	VGP2OS2 (P2OS2)	VGP3OS2 (P3OS2)
VisualAge Generator Middleware using LU62	VGS2OS2 (S2OS2)	VGS3OS2 (S3OS2)

- Display the following control and performance information:

Response time

In seconds, the time elapsed between the issuance of the call by the GUI client and the return from a two- or three-tier server call

EZERT8 Any error code information generated by a call from a GUI to the two-tier server and a call from a two-tier server to a three-tier server

EZEUSRID

User ID value as seen by the two-tier and three-tier servers.

Figure 120 shows the primary window of the VGPING GUI client application.

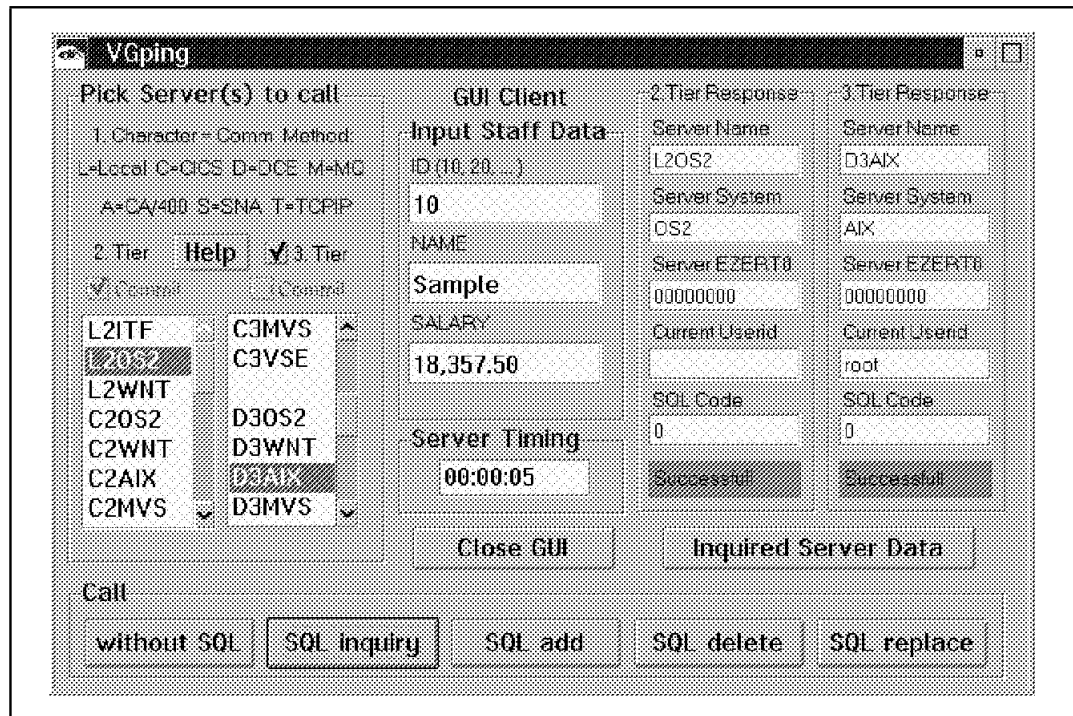


Figure 120. Primary Window of VGPING Sample Application Client

Note: We began to provide support for a choice of client- or server-based LUW management based on the LUW toggle button setting, but chose instead to hard-code a server-based LUW management approach.

- Display SQL data retrieved by each server for both two- and three-tier configurations. Data for the tier-2 server is displayed on the primary window. All SQL data retrieved by both the tier-2 and tier-3 servers can be displayed by clicking on the **Inquired Server Data** push button to open the server data window (see Figure 121 on page 167).

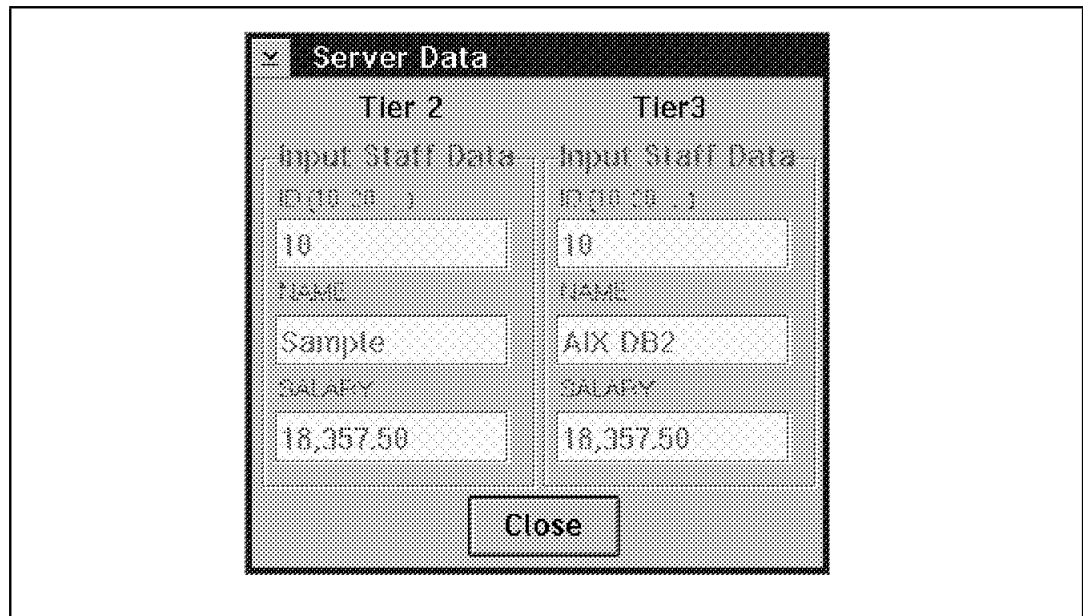


Figure 121. Server Data Window for VGPING Sample Application Client

There are many uniquely named server applications, representing most of the available two- and three-tier client/server configurations supported by VisualAge Generator. Each server application for a given tier has the same main process. Figure 122 shows the structure diagram of one of the tier-2 servers.

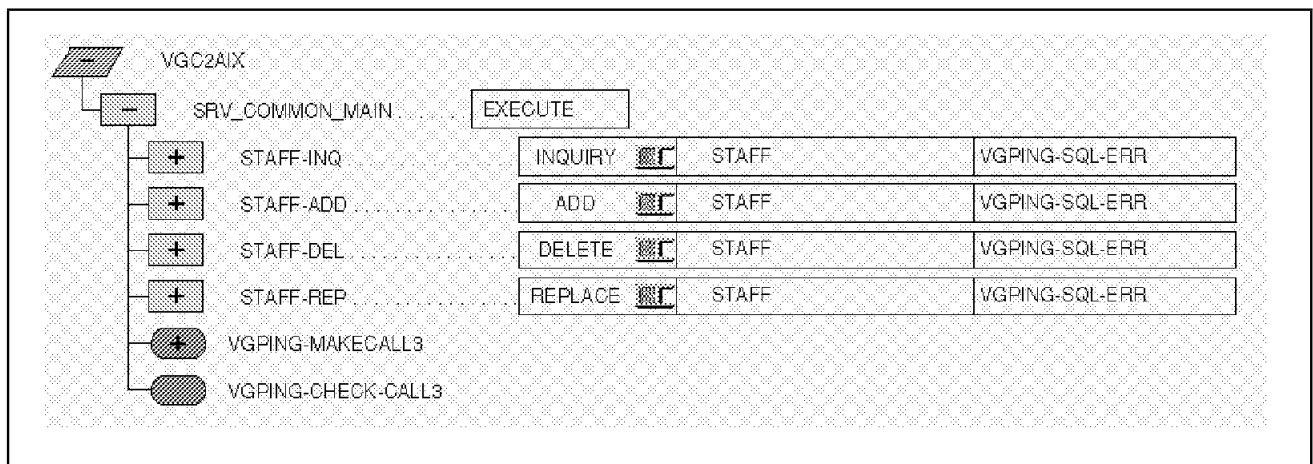


Figure 122. Structure of Tier-2 Server Application

The tier-3 server structure is similar, but does not support calling additional server applications.

A.1.4 Sample Application Materials

The following materials are provided on the diskette included with this publication. Refer to the Readme file on the diskette for details.

Sample application source

The complete sample application is provided in ESF format.

Sample database

Each server application accesses an SQL table named STAFF. The STAFF table was implemented in the DB2 environment used in each target run time environment.

The STAFF table is found in the sample database provided as part of most DB2 database installations. An IXF file for this table is provided on the diskette.

Generation command file

The sample application system was generated and prepared as required for each target runtime environment.

The generation command file for the sample application system is provided on the diskette.

Linkage table

The linkage table files used during generation and runtime processing are provided on the diskette.

A.2 DUOW Sample Application

To provide support for testing multiple remote and distributed database access configurations, we designed and implemented the DUOW sample application.

A.2.1 Objectives

The objective of the DUOW sample application was to make our job of building and testing different remote and distributed database access configurations easier. To do this, we built a custom VisualAge Generator application that satisfies these functional objectives:

- One GUI application serves for all possible database configurations.
- Supports remote and distributed database access configurations.
- Provides default SQL support on the target database platforms.
- Allows ITF-based or generated execution of the sample application server.
- Captures VisualAge Generator provided return codes for server SQL requests.
- Measures elapsed time for the server call.

A.2.2 Overview

Our DUOW sample application can access a table in one or two different databases. We use the EZECONCT statement to implement single database access using a remote database access connection and multiple database access using the available DUOW features.

The table name used in each database is the same: STAFF. The creator ID for the table is based on:

- The user ID used during SQL bind processing for remote database access
- The user ID used during EZECONCT processing for distributed database access

This allows us to violate one of the recommended DUOW design guidelines:

Put SQL statements that access different databases in different applications. An application cannot be bound to a database unless all tables referenced in the application are in the database.

Because our table name is the same in each database we connect to, we can bind the VisualAge Generator application to each target database without problems.

A.2.3 Function

The DUOW sample application provides the following functions:

- Selection of one or more target databases for SQL processing
- Identification of user ID and password information to be used on access using the DUOW features of the EZECONCT statement
- Push button selection of the processing request:

Select, Update, Delete, Insert

Call servers with SQL select processing using key and associated data defined in the VGPING client user interface.

SQL requests will either be issued against the STAFF table in a single database using remote database access or against the STAFF table in two databases using DUOW support. The notebook page active at the time of the request determines which database will be used: one or both.

- Display the following control and performance information:

Response time	In seconds, the time elapsed between issuance of a call by the GUI client and the return from database logic server
EZERSQLCD	Any error code information generated by SQL requests issued by the database logic server

Our DUOW sample is implemented as a GUI application (DUOWGUI) and a called server application (DUOWAPP). Figure 123 on page 170 shows the primary window of the DUOW GUI client application.

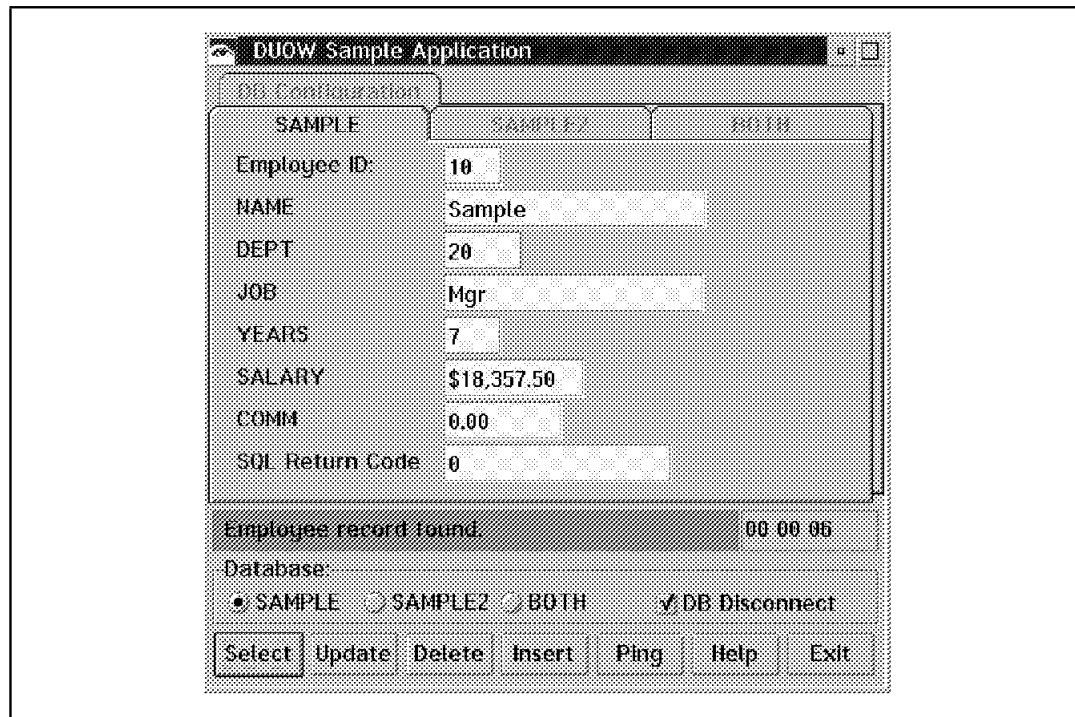


Figure 123. Primary Window for DUOW Sample Application Client

The actual database connection and access processing is implemented in the called application DUOWAPP. Figure 124 shows the structure diagram and the main process of the DUOWAPP application.

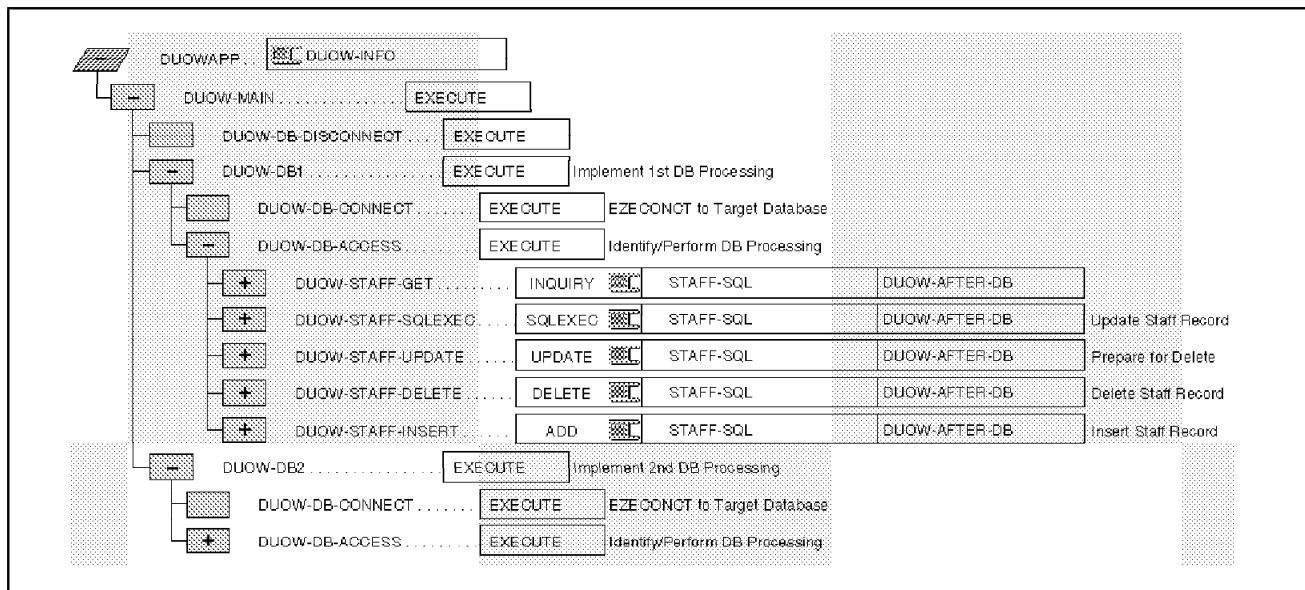


Figure 124. Structure of DUOW Sample Application Server

A.2.4 Sample Application Materials

The following materials are provided on the diskette included with this publication. Refer to the Readme file on the diskette for details.

Sample application source

The complete sample application is provided in ESF format.

Sample database

Each server application accesses an SQL table named STAFF. The STAFF table was implemented in the DB2 environment used in each target runtime environment.

The STAFF table is found in the sample database provided as part of most DB2 database installations. An IXF file for this table is provided on the diskette.

Generation command file

The sample application system was generated and prepared as required for each target runtime environment.

The generation command file for the sample application system is provided on the diskette.

Linkage table

The linkage table files used during generation and runtime processing are provided on the diskette.

Appendix B. Special Notices

This publication is intended to help VisualAge Generator application programmers and system administrators understand the available VisualAge Generator client/server implementation options and configure a working system using either a CICS-, DCE-, or database-enabled client/server configuration. The information in this publication is not intended as the specification of any programming interfaces that are provided by VisualAge Generator Developer or VisualAge Generator Workgroup Services. See the PUBLICATIONS section of the IBM Programming Announcement for VisualAge Generator Developer or VisualAge Generator Workgroup Services for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
CICS/ESA
CICS/VSE
DB2
DB2/6000

CICS OS/2
CICS/MVS
CICS/6000
DB2/2
DRDA

IBM
Open Blueprint
VisualAge
VSE/ESA

MVS/ESA
OS/2
VisualGen
VTAM

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other trademarks are trademarks of their respective companies.

Appendix C. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 177:

- *Implementing VisualGen Client/Server Communication*, GG24-4235
- *CICS Clients Unmasked*, SG24-2534-01
- *Understanding OSF DCE 1.1 for AIX and OS/2*, SG24-4616
- *DCE Cell Design Considerations*, SG24-4746
- *Distributed Relational Database Cross Platform Connectivity and Applications*, SG24-4311
- *DB2 for MVS Connections with AIX and OS/2*, SG24-4558

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

C.3 Other Publications

These publications are also relevant as further information sources:

- *Introduction to the Open Blueprint: A Guide to Distributed Computing*, G326-0395-01
- *Open Blueprint Technical Overview*, GC23-3808-01
- *Running VisualAge Generator Applications on OS/2, AIX, and Windows*, SH23-0235-00
- *Generating VisualAge Generator Applications*, SH23-0227-00
- *Designing and Developing VisualAge Generator Applications*, SH23-0228-00
- *Developing VisualAge Generator Client/Server Applications*, SH23-0230-00
- *Installing VisualAge Generator Workgroup Services*, GH23-0240-00
- *Installing and Using OS/2 Clients*, S20H-4782
- *DB2 Administration Guide*, S20H-4580

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL <http://www.redbooks.ibm.com>.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type `GOPHER.WTSCPOK.ITSO.IBM.COM`
- **Tools disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get BookManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
```

To register for information on workshops, residencies, and redbooks:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
```

For a list of product area specialists in the ITSO:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Home Page on the World Wide Web**
<http://w3.itso.ibm.com/redbooks>
- **IBM Direct Publications Catalog on the World Wide Web**
<http://www.elink.ibm.link.ibm.com/pb1/pb1>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	Internet
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1)001-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Home Page <http://www.redbooks.ibm.com>
IBM Direct Publications Catalog <http://www.elink.ibm.link.ibm.com/pbl/pbl>

- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number
• Invoice to customer number		
• Credit card number		

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

ACL	access control list	ECI	external call interface
ADSM	ADSTAR Distributed Storage Manager	ESF	external source format
API	application program interface	GA	general availability
APPC	advanced program-to-program communication	GUI	graphical user interface
APPN	advanced peer-to-peer networking	ID	identification
ASCII	American National Standard Code for Information Interchange	IMS	Information Management System
ATM	asynchronous transfer mode	IPX	Internetwork Packet eXchange
CASE	computer assisted software engineering	ITF	interactive test facility
CDS	Cell Directory Service (OSF/DCE)	IXF	integrated exchange format
CEDA	resource definition online transaction (CICS)	LAN	local area network
CHS	Simplified Chinese	LD	listener definition
CICS	Customer Information Control System	LU	logical unit
CPMI	default ECI program transaction (CICS)	LUW	logical unit of work
DBCS	double-byte character set	MIT	Massachusetts Institute of Technology
DCE	Distributed Computing Environment (OSF)	MSL	member specification library
DFSMS	Data Facility Storage Management Subsystem (MVS and VM)	MVS	Multiple Virtual Storage
DLC	data link control	NDF	network definition file
DDL	data definition language	NetBIOS	Network Basic Input/Output System
DOUW	distributed unit of work	NLS	national language support
DPL	distributed program link	OSI	open systems interconnection
DRDA	Distributed Relational Database Architecture	PCT	program control table
DSS	directory and security server	POSIX	Portable Operating System Interface for Computer Environments
DTS	distributed time service	PPT	processing program table
EBCDIC	extended binary-coded decimal interchange code	RACF	Resource Access Control Facility
		RCT	resource definition on-line
		REXEC	remote execution

<i>RPC</i>	remote procedure call	<i>TCS</i>	connection and session table
<i>RU</i>	request/response unit	<i>TCT</i>	terminal control table
<i>SDK</i>	software developers kit	<i>TM</i>	transaction manager
<i>SI</i>	shift in	<i>TSO</i>	Time Sharing Option
<i>SIT</i>	system initialization table	<i>TUI</i>	text user interface
<i>SNA</i>	System Network Architecture	<i>TWA</i>	transaction work area
<i>SNT</i>	sign-on table	<i>UPM</i>	User Profile Manager
<i>SO</i>	shift out	<i>VSE</i>	Virtual Storage Extended
<i>SQL</i>	Structured Query Language	<i>WAN</i>	wide area network
<i>TCP/IP</i>	transfer control protocol/Internet protocol		

Index

A

- abbreviations 181
- access control lists
 - authenticated 64
 - checking current entries 89
 - description 61
 - group authority 63
 - initial container creation 61
 - initial object creation 61
 - object 61
 - removing unauthenticated 90
 - security 89
 - target directory 65
 - test authority 64
 - unauthenticated 64
- acronyms 181
- application server workstation
 - description 57
 - implementing multiple workstations 84
 - OS/2 configuration 65
 - Windows NT configuration 70

B

- bibliography 175

C

- CDS client
 - description 57
- CICS Client
 - client/server communication support 44
 - making CICS connections 44, 152
- CICS OS/2
 - CAIM transaction 34
 - CEDA transaction 39, 154
 - CICS Client connections 152
 - cold start 36
 - Communications Manager/2 149
 - configuration 29, 30
 - connecting CICS servers 49, 149
 - CPMI transaction 38
 - customization 30
 - DBCS customization 147

CICS OS/2 (*continued*)

- DBCS implementation 154
- DFHMIR 38
- ECI 38
- ELAENV.CMD 33
- ELAEX300.BTR 34
- ELARUNC.CMD 33
- FAAAEFIE.BTR 34
- IBM COBOL 34
- NetBIOS 31
- protocols 31
- security 31
- SNT 31
- software 30
- starting with Workgroup Services 34
- system initialization table 30, 32, 154
- TCP/IP 31
- transaction definition 38
- transaction work area 36
- UPM 31
- VisualAge COBOL 34
- Workgroup Services configuration 33

- CICS SYNCPOINT 10
- CICS SYNCPOINT CANCEL 10
- CICS-based client/server
 - CICS Client connections 152
 - CICS OS/2 29
 - CICS/6000 39
 - clients 28
 - configuration 25, 26
 - connecting CICS servers 49, 149
 - ITF as client 49
 - options 25
 - processing flow 27
 - protocol choices 26
 - scenarios 28
 - servers 28
 - tracing calls 48
- CICS-based client/server communication
 - introduction 9
 - Open Blueprint assessment 9
- CICS/6000
 - AIX user customization 42
 - configuration 39, 40
 - connecting CICS servers 49

- CICS/6000 (*continued*)
 - customization 40, 41
 - DB2 shared object 42
 - defining programs 43
 - environment variables 41
 - listener definition 40
 - operations 44
 - restarting 44
 - REXEC 42
 - shutting down 44
 - software 39
 - TCP/IP 40
 - Workgroup Services 41
 - Workgroup Services installation 39
- client/server
 - distributed data 103
 - distributed function 103
 - distributed presentation 103
 - implementation issues 16
 - overview 103
 - remote data 103
 - remote presentation 103
- cold start 34, 36
- communication services
 - capabilities 8
 - CICS-based 9
 - DCE-based 9
 - description 5
 - DRDA 8
 - introduction 3
 - message queuing 8
 - options 8
 - protocols 8
 - remote procedure call 8
 - VisualAge Generator middleware 9
- Communications Manager/2
 - DBCS configuration 149
 - DRDA connections
 - network definition file 120
 - network definition file 120
- configuration
 - CICS 25
 - CICS client 44
 - CICS OS/2 29
 - CICS/6000 39
 - connecting CICS servers 49
 - CSODCES 75
 - DB2 116
 - DBCS-enabled client/server 139
 - DCE 53
 - DCE.CNF 77
 - DRDA connections 120
 - overview 139
 - starting BookManager 140
- CSODCES
 - CDS objects 79
 - configuration 75
 - configuration file 75

- CSODCES (*continued*)
 - DCE.CNF configuration file 77
 - DCEACLObject 76
 - DCEprincipal 76
 - key table 77
 - parameters 84
 - PUBLIC 76
 - rgy_edit 77
 - SECURE 76
 - server principal authority 62
 - SERVERID 76
 - starting 77, 78
 - startup options 84
 - user definitions 59
- CSOLINKTBL 20
- CSOTROPT 48
- CSOUEXIT 22, 23

D

- database
 - authorization 21, 22, 23, 37
 - CICS OS/2 36
 - CSOUEXIT 22, 23
 - DB2 shared object 42
 - environment variables 21, 22
 - identification 21, 22, 36
 - logon 22
 - testing 21
 - UPM 22, 23
- database-enabled client/server
 - bind 126, 131, 133
 - common server connections 117
 - configuration 116
 - DB2 common server connectivity 104
 - options 104
 - remote access 107
 - configuration options
 - DataJoiner 105
 - DRDA connectivity 104
- database
 - authorization 107
 - common server connections 117
 - connection 107
 - DRDA connections 120
 - DUOW 111
 - DUOW configurations 116
 - identification 108
 - implicit connection 109
 - logon 107
 - multiple database connection 114
 - sample application 116
 - testing with ITF 115
 - transaction manager database 113
- DRDA connections
 - configuration 120
 - network definition file 120
- DUOW configurations 116

- database-enabled client/server (*continued*)
 - overview 103
 - service name 118
 - testing
 - bind 126, 131, 133
 - DUOW access 126
 - VisualAge Generator support 106
- DataJoiner 105
- DB2
 - bind 126, 131, 133
 - common server connections 117
 - common server connectivity 104
 - configuration 116
 - remote access 107
 - database
 - authorization 107
 - common server connections 117
 - connection 107
 - DRDA connections 120
 - DUOW 111
 - DUOW configurations 116
 - identification 108
 - implicit connection 109
 - logon 107
 - multiple database connection 114
 - sample application 116
 - testing with ITF 115
 - transaction manager database 113
- DataJoiner 105
- DRDA connections
 - configuration 120
 - network definition file 120
- DRDA connectivity 104
- DUOW configurations 116
- service name 118
- testing
 - bind 126, 131, 133
 - DUOW access 126
- VisualAge Generator database access 106
- DB2DBDFT 22
- DBCS-enabled client/server
 - CICS OS/2 customization 147
 - Communications Manager/2 149
 - configurations 139
 - customization 145
 - data definition 140
 - environment setup 145
 - generation 147
 - hardware and software used 142
 - IBM COBOL 145
 - implementation 141
 - introduction 139
 - linkage table definitions 148
 - preparation 147
 - relational database 141
 - sample application 143, 147
 - scenarios 141
 - three-tier 141

- DBCS-enabled client/server (*continued*)
 - two-tier 141
 - VisualAge COBOL 145
 - VisualAge Generator configuration 145, 146
- DCE cell
 - description 57
- DCE client
 - description 57
- DCE server program
 - CDS objects 79
 - configuration 75
 - DCE.CNF configuration file 77
 - description 57
 - key table 77
 - rgy_edit 77
 - starting 77, 78
- DCE-based client/server
 - access control lists
 - authenticated 64
 - description 61
 - group authority 63
 - initial container creation 61
 - initial object creation 61
 - object 61
 - target directory 65
 - test authority 64
 - unauthenticated 64
- AIX
 - installation 59
 - management 60
- application server workstation
 - implementing multiple workstations 84
 - OS/2 configuration 65
 - Windows NT configuration 70
- authenticated 64
- cell server 59
- clients 56
- configuration
 - application server workstation 65, 70
 - authorization required 88
 - basics 56
 - GUI client workstation 68, 73
 - logon required 88
 - options 53
 - OS/2 65, 68
 - OS/2 DCE client 65
 - OS/2 DCE slim client 68
 - overview 57
 - protocol choices 54
 - scenario 57
 - security 88
 - tasks 60
 - Windows 95 73
 - Windows 95 DCE client 73
 - Windows NT 70
 - Windows NT DCE client 70
- creating directories 61, 65
- dcecp
 - adding principal to group 62

DCE-based client/server (*continued*)

dcecp (*continued*)

- directories 61, 65
- groups 60
- organizations 60
- server principal authority 62
- target directory 65
- users 60

dcesecure 88

GUI client workstation

- OS/2 configuration 68
- Windows 95 configuration 73

installation

- AIX 59

management 60

processing flow 55

rgy_edit 77

- key table 77

running applications

- DCE server program configuration 75

- DCE.CNF configuration file 77

- overview 75

- starting DCE server program 77, 78

- trace log 81

- using the sample application 80

scenarios 56

security

- authorization required 88

- dcesecure 88

- forcing logon 89

- implementation 88

- implementing authorization checking 93

- logon required 88

- unauthenticated 88

- using dcesecure 93

server principal authority 62

servers 56

slim client 68

target directory 65

TCP/IP 58

terminology

- application server workstation 57

- CDS client 57

- DCE cell 57

- DCE client 57

- DCE server program 57

- GUI client workstation 57

- VisualAge Generator server application 57

unauthenticated 64

user definitions

- client 59

- CSODCES 59

- dcecp 60

- group 59

- implementation 60

- organization 59

- principals 59

- REMOTECOMTYPE=dcesecure 59

- server 59

DCE-based client/server communication

- introduction 9

- Open Blueprint assessment 12

DCE.CNF 77

dcecp

- adding principal to group 62

- directories 61, 65

- groups 60

- organizations 60

- server principal authority 62

- target directory 65

- users 60

DFHMIR 38

distributed data 103

distributed function 103

distributed presentation 103

DPATH

- linkage table 20

DRDA 8

- connectivity 104

DUOW

- EZECONNECT 111

- implementation 111

- multiple database connection 114

- testing with ITF 115

- transaction manager database 113

E

ELAENV.CMD 33

ELARTRDB_tttt 22

ELARUNC.CMD 33

environment variables

- CICS/6000 41

- CSOLINKTBL 78, 145

- CSOTIMEOUT 145

- CSOTROPT 48, 78, 145

- CSOTROUT 145

- CSOUEXIT 22, 23, 145

- DB2DBDFT 22, 108

- DB2INSTANCE 41

- DBCS-enabled client/server 145

- ELARTRDB_tttt 22, 36, 108

- EZERNLS 41

- EZERSQLDB 22, 36, 41, 78, 108

- FCWDB2DIR 41

- FCWDBNAME_appl 22, 108

- FCWDBPASSWORD 41

- FCWDBUSER 41

- FCWDPATH 41

- FCWLIBPATH 41

- FCWTROPT 78, 109

- LIBPATH 41

- SQLDBDFT 108

- working with 108

errors

- CAIM import failure 35

- CSO7800E 100

errors (*continued*)

- CSO7801E 101
- DCE setup and configuration 98
- DCE-based client/server 98
- EZE4240E 38
- FAA5513E 34
- LNK1083 38
- SQL 98
- SQL0428N 136
- SQL0551N 98, 130, 136
- SQL0567N 136
- SQL0805N 136
- SQL0818N 136
- SQL0859N 136
- SQL0868N 136
- SQL0900N 136
- SQL1013N 136
- SQL1024N 136
- SQL1246N 129, 136
- SQL1403N 136
- SQL30082N 136
- SQL30090N 136
- EZECOMIT 10
- EZCONNECT
 - DUOW 111
 - using 111
- EZEROLLB 10
- EZERSQLDB 22

F

- FCWDBNAME_appl 22

G

- global transparent access 7
- GUI client workstation
 - description 57
 - OS/2 configuration 68
 - Windows 95 configuration 73

I

- IBM COBOL
 - CICS configuration 34
 - DBCS-enabled client/server 145
- integration
 - global transparent access 7
 - network-wide directory 7
 - network-wide security 7
 - single sign-on 7
- introduction
 - client/server configuration options 2
 - communication services 3
 - integration objectives 7
 - objectives 2
 - Open Blueprint 3

introduction (*continued*)

- project scope 2
- ITF 19

K

- key table 77

L

- linkage table
 - :CALLLINK 19, 20
 - CICS OS/2 servers 37
 - CSOLINKTBL 20
 - DCE OS/2 client 69
 - DCE OS/2 servers 67
 - DCE Windows 95 client 74
 - DCE Windows NT servers 71
 - determining active 20
 - DPATH 20
 - external calls 20
 - LINKTYPE 20
 - remote calls 19, 20
- listener definition 40

M

- message queuing 8
- MVS CICS
 - CEDA transaction 157
 - CICS Client connections 152, 157
 - Communications Manager/2 149
 - connecting CICS servers 149
 - connection definitions 157
 - DBCS implementation 157
 - mirror transaction definition 160
 - program definition 162
 - session definitions 158

N

- NetBIOS 31
- network-wide directory 7
- network-wide security 7
- NLS 140

O

- objectives 2
- Open Blueprint
 - assessment 9, 12, 13, 16
 - CICS-based client/server communication 9
 - DCE-based client/server communication 12
 - summary 16
 - VisualAge Generator middleware client/server communication 13

Open Blueprint (*continued*)

- communication services 3
- components 4
 - application-enabling services 4
 - Application/Workgroup services 4
 - applications and development tools 4
 - common transport semantics 5
 - communication services 5
 - data access services 5
 - distributed-systems services 5
 - distribution services 5
 - local operating system services 5
 - network services 5
 - presentation services 4
 - signalling and control plane 5
 - subnetworking 5
 - systems management 5
- critical services
 - application/workgroup services 5
 - distribution services 6
 - naming and directory 6
 - security 6
 - time 6
 - transaction manager 7
 - transaction monitor 5
- introduction 3
- positioning VisualAge Generator 3

P

- PING 58
- principals 59
- project scope 2
- protocols 8
 - choices 26, 54
 - CICS OS/2 31
 - CICS-based client/server 26
 - DCE-based client/server 54
 - NetBIOS 31
 - TCP/IP 31, 40

R

- READIBM.EXE 140
- remote data 103
- remote presentation 103
- remote procedure call
 - implementation options 9
 - introduction 8
- REXEC 42
- rgy_edit 77
- runtime
 - CSOUEXIT 23
 - database authorization 23
 - database identification 22
 - environment variables 22
 - UPM 23

S

- sample application
 - DBCS-enabled client/server 143
 - description 163
 - DUOW configurations 116
 - generation
 - CICS OS/2 servers 37, 38
 - CICS/6000 servers 41, 43
 - DCE OS/2 client 69
 - DCE OS/2 servers 67
 - DCE Windows 95 client 74
 - DCE Windows NT servers 72
 - GUI client for CICS servers 45
 - introduction 163
 - linkage table
 - CICS OS/2 servers 37
 - DCE OS/2 client 69
 - DCE OS/2 servers 67
 - DCE Windows 95 client 74
 - DCE Windows NT servers 71
 - GUI client generation for CICS 46
 - GUI client runtime for CICS 48
 - preparation
 - CICS OS/2 servers 37
 - CICS/6000 servers 41
 - running
 - DCE trace log 81
 - DCE-based client/server 80
 - VGPING
 - description 163, 168
 - function 165, 169
 - introduction 163, 168
 - objectives 163, 168
 - overview 164, 168
 - program materials 167, 171
 - user interface 166, 169
- security
 - CICS OS/2 31
 - DCE-based client/server
 - authorization required 88
 - forcing logon 89
 - implementation 88
 - implementing authorization checking 93
 - logon required 88
 - using dcesecure 93
 - dcesecure 88
 - SNT 31
 - unauthenticated 88
 - UPM 31
- service name 118
- single sign-on 7
- SNT 31
- SQL COMMIT WORK 10
- SQL ROLLBACK 10
- system initialization table
 - CICS OS/2 30, 32, 36, 154
 - DBCS implementation 154
- examples 32

system initialization table (*continued*)
transaction work area 36

VisualAge Generator server application
description 57

T

TCP/IP

CICS OS/2 31
CICS/6000 40
DB2 configuration 117
DBCS client/server communication 141
DCE 58
PING 58
service name 118

testing

:CALLLINK 19
bind 126, 131, 133
calling generated applications 19
CSOLINKTBL 20
CSOUEXIT 22
current ID 116
database authorization 21, 22
database identification 21
DUOW access
bind 126, 131, 133
generation 126
preparation 126
with local and remote databases 131
with local databases 127
with remote databases 133
DUOW with ITF 115
external calls 20
EZECONCT 115
ITF 19
linkage table 19, 20
remote calls 20
UPM 22

U

UPM

CICS OS/2 31
database authorization 22, 23

V

VisualAge COBOL

CICS configuration 34
DBCS-enabled client/server 145

VisualAge Generator

database
identification 108

VisualAge Generator middleware client/server communication

implementation issues 16
introduction 9
Open Blueprint assessment 13

W

Workgroup Services

CAIM transaction 34
CICS configuration 33, 34, 36
CICS/6000 41
CICS/6000 installation 39
ELAENV.CMD 33
ELAEX300.BTR 34
ELAM transaction 36
ELARUNC.CMD 33
FAAAEFIE.BTR 34
transaction work area 36

ITSO Redbook Evaluation

VisualAge Generator Client/Server Communications Examining the Options
SG24-4237-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redeval@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)



Printed in U.S.A.

SG24-4237-00

